T.C.
ISTANBUL AYDIN UNIVERSITY
INSTITUTE OF GRADUATE STUDIES

ENHANCING WEB ACCESSIBILITY USING DEEP CONVOLUTIONAL NETWORKS AND NATURAL LANGUAGE PROCESSING TECHNIQUES

MASTER'S THESIS

Muhammad Kashif Shaikh

Department of Software Engineering

Artificial Intelligence and Data Science Program

June, 2023

T.C.
ISTANBUL AYDIN UNIVERSITY
INSTITUTE OF GRADUATE STUDIES

ENHANCING WEB ACCESSIBILITY USING DEEP CONVOLUTIONAL NETWORKS AND NATURAL LANGUAGE PROCESSING TECHNIQUES

MASTER'S THESIS

Muhammad Kashif Shaikh

(Y2013.140016)

Department of Software Engineering

Artificial Intelligence and Data Science Program

Thesis Advisor: Assist. Prof. Dr. JAWAD RASHEED

June, 2023

APPROVAL FORM

DECLARATION

I hereby declare with respect that the study " Enhancing Web Accessibility Using Deep Convolutional Networks And Natural Language Processing Techniques", which I submitted as a Master thesis, is written without any assistance in violation of scientific ethics and traditions in all the processes from the Project phase to the conclusion of the thesis and that the works I have benefited are from those shown in the Bibliography. (…/…/2023)

Muhammad Kashif Shaikh

FOREWORD

I would like to thank Almighty Allah for letting me achieve my goals, without His mercy I wouldn’t have come this far.

I would like to take this opportunity to express my deepest gratitude to several individuals who have made this thesis possible. Firstly, I would like to thank my thesis supervisor Assist. Prof. Dr. JAWAD RASHEED for their unwavering support, guidance, and expertise throughout the research process. Their invaluable feedback and encouragement have been instrumental in shaping this work.

I would also like to express my gratitude to the faculty members in the Department of Software Engineering for providing me with a challenging and stimulating academic environment that has fostered my intellectual growth.

Furthermore, I extend my appreciation to my family and friends for their unconditional love, encouragement, and support throughout my academic journey. They have been my pillars of strength and motivation, and I am deeply grateful for their unwavering support.

Finally, I would like to acknowledge the invaluable contributions of all the participants who participated in this study, without whom this research would not have been possible.

Thank you all for your support and encouragement.

ENHANCING WEB ACCESSIBILITY USING DEEP CONVOLUTIONAL NETWORKS AND NATURAL LANGUAGE TECHNIQUES

ABSTRACT

Deep neural networks (DNN) and Convolutional neural networks (CNN) are artificial neural networks used for image classification, natural language processing, object detection, and image segmentation. These techniques aid in the friendly usage of websites for people who have some kind of disability making it difficult for them to access. In this study, DNN and CNN were opted and employed to generate captions for the given images using different datasets, and metrics such as, BLEU and WER were used for system evaluation. The study's results revealed promising outcomes, highlighting the efficacy of deep learning techniques in enhancing web accessibility for individuals with visual impairments. The developed system effectively enhances the browsing experience and improves information accessibility for individuals with print impairments by providing precise and descriptive captions for images. These advancements align with the broader objective of enabling intelligent machines through the utilization of natural language processing (NLP) and facilitating linguistic-based communication between humans and computers.

Keywords: CNN, Attention model, BLEU, WER, DenseNet

ENHANCING WEB ACCESSIBILITY USING DEEP CONVOLUTIONAL NETWORKS AND NATURAL LANGUAGE TECHNIQUES

ÖZET

Derin sinir ağları (DNN) ve Evrişimli sinir ağları (CNN), görüntü sınıflandırma, doğal dil işleme, nesne algılama ve görüntü bölümleme için kullanılan yapay sinir ağlarıdır. Bu teknikler, web sitelerinin erişimini zorlaştıran bir tür engeli olan kişiler için web sitelerinin dostça kullanımına yardımcı olur. Bu çalışmada, farklı veri kümeleri kullanılarak verilen görüntüler için altyazı oluşturmak için DNN ve CNN seçilmiş ve kullanılmış ve sistem değerlendirmesi için BLEU ve WER gibi metrikler kullanılmıştır. Çalışmanın sonuçları, derin öğrenme tekniklerinin görme engelli bireyler için web erişilebilirliğini artırmadaki etkinliğini vurgulayarak umut verici sonuçlar ortaya koydu. Etkili bir şekilde geliştirilmiş sistem, resimler için kesin ve açıklayıcı altyazılar sağlayarak, tarama deneyimini geliştirir ve baskı bozukluğu olan kişiler için bilgiye erişilebilirliği geliştirir. Bu gelişmeler, doğal dil işlemenin (NLP) kullanımı yoluyla akıllı makinelere olanak sağlama ve insanlar ile bilgisayarlar arasında dil tabanlı iletişimi kolaylaştırma gibi daha geniş bir hedefle uyumludur.

Anahtar Kelimeler: CNN, Attention model, BLEU, WER, DenseNet

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

ADA Americans with Disabilities Act

API Application programming interface

ATAG Authoring Tool Accessibility Guidelines

BLEU BiLingual Evaluation Understudy

CNN Convolutional neural network

CRPD Committee on the Rights of Persons with Disabilities

HTML Hyper Text Markup Language

LSTM Long short-term memory

UAAG User Agent Accessibility Guidelines

W3C World Wide Web Consortium

WAI Web Accessibility Initiative

WCAG Web Content Accessibility Guidelines

WER Word Error Rate

LIST OF TABLES

LİST OF FİGURES

I. INTRODUCTION

A. ACCESSIBILITY AND WEB ACCESSIBILITY

"The power of the Web is in its universality. Access by everyone regardless of disability is an essential aspect" (Henry).

Accessibility can be defined as those individuals having any type of disability have right to live a normal life like any other person who hasn't any disability. The evolution of accessibility has encompassed various perspectives on disabilities, disability rights activism leading to the establishment of laws and policies, and the impact of technology. But it can be observed that technology can both create and address accessibility challenges. In 2007, a treaty was signed under United Nation's Convention emphasizing the Rights of Persons with Disabilities (CRPD) which particularly was "intended to protect the rights and dignity of people with disabilities". To enable persons with disabilities to live independently and participate fully in all aspects of life, States Parties shall take appropriate measures to ensure to persons with disabilities access, on an equal basis with others, to the physical environment, to transportation, to information and communications, including information and communications technologies and systems, and to other facilities and services open or provided to the public, both in urban and in rural areas", says the Article 9, Chapter 3, Web accessibility 21 (Jokinen 2020)

Literature reveals that web is hardly three decades old, followed by the concept of web accessibility, which is considered as a subcategory of computer accessibility, hence minimizing the hinderance in terms of computer usage. In 1984, a study was conducted in order to investigate the efforts people with disabilities had to make while using computer software and hardware also (Bowe and Little 1984). The concept of web accessibility pertains to the capacity to obtain and engage with web pages irrespective of any disabilities or impairments that may be present in any individual, which can be fruitful only when it can be implemented without any hinderance, easy to grasp, user-friendly and the matter in it is understandable (Jokinen 2020). Creating an accessible website involves designing it in a way that enables users including those with disabilities like visual impairment, to navigate and interact with it easily and effectively via electronic devices is called web accessibility which aids in facilitating the utilization, comprehension, navigation, and interaction with web content by people with disabilities by mitigating the effects of various disabilities that hinder Internet access, thereby (Martínez, De Andrés et al. 2014). It can be observed that most of the websites are inaccessible or semi-accessible (Hashemian 2011, Nahon, Benbasat et al. 2012). Some of the web accessibility related limitations are as follows (López, Pascual et al. 2011, Baowaly and Bhuiyan 2012, Tuan and Phan 2012, Brown and Hollier 2015): a) not having enough knowledge regarding web accessibility, its designing and implementation, b) having limited resources to deal with accessibility problems, c) relevant professional personnel having expertise in accessibility evaluation tools, d) not providing relevant manuals and training sessions (Abuaddous, Jali et al. 2016).

B. WEB CONTENT ACCESSIBILITY GUIDELINES

Although many guideline standards have been developed but the Web Content Accessibility Guidelines (WCAG) from Web Accessibility Initiative (WAI) of the World Wide Web Consortium (W3C) is considered to be the most implemented guidelines. the Section 508 of the US Rehabilitation Act (US Government 2011) and the Web Accessibility Code of Practice published by the British Standard Institute (British Standards Institute 2010) are the other guideline sets proposed by government of respective countries. WAI developed accessibility model which has three sets of guidelines, namely, the User Agent Accessibility Guidelines (UAAG) (Jacobs et al. 2002), the Web Content Accessibility Guidelines (WCAG) (Chisholm et al. 1999, Caldwell et al. 2008) and the Authoring Tool Accessibility Guidelines (ATAG) (Treviranus et al. 2000) for those professionals which develop authoring tools, web content and other web browsing and assistive technologies also, aiming of promoting web accessibility for individuals with disabilities. It is expected that adherence to these guidelines by web content developers employing compatible authoring tools, and rendered by compatible user agents will enhance website accessibility for users with disabilities. WAI also developed the Web Content Accessibility Guidelines (WCAG) to promote the developers to develop websites with accessible content. There are three versions of WCAG, i.e., WCAG 1.0, WCAG 2.0 and WCAG 3.0, where Version 1 was developed in 1999, version 2 in 2008 and version 3 in 2021 (Chisholm, Vanderheiden et al. 2001, Caldwell, Cooper et al. 2008, Freire 2012) Version 2.1 was released in 2018. Many countries such as Germany, United Kingdom, Brazil, France, Japan, Canada, Italy, Australia, Chile, Honduras, the Netherlands, Portugal, and South Korea, have developed their own regulations. Other guidelines were also proposed by WAI are WAI-ARIA – Accessible Rich Internet Applications. WCAG 2.0 is structured based on four fundamental design principles, namely perceivable, operable, understandable, and robust, that serve as the bedrock for ensuring web accessibility.Within these principles, there exist twelve guidelines, each of which is associated with one or more testable success criteria (SCs). The SCs are classified into three levels: A (lowest), AA (medium), and AAA (highest), totaling to 61 SCs. It is worth noting that a single accessibility issue can be addressed by more than one SC at different levels of priority within WCAG 2.0 (Abuaddous, Jali et al. 2016).

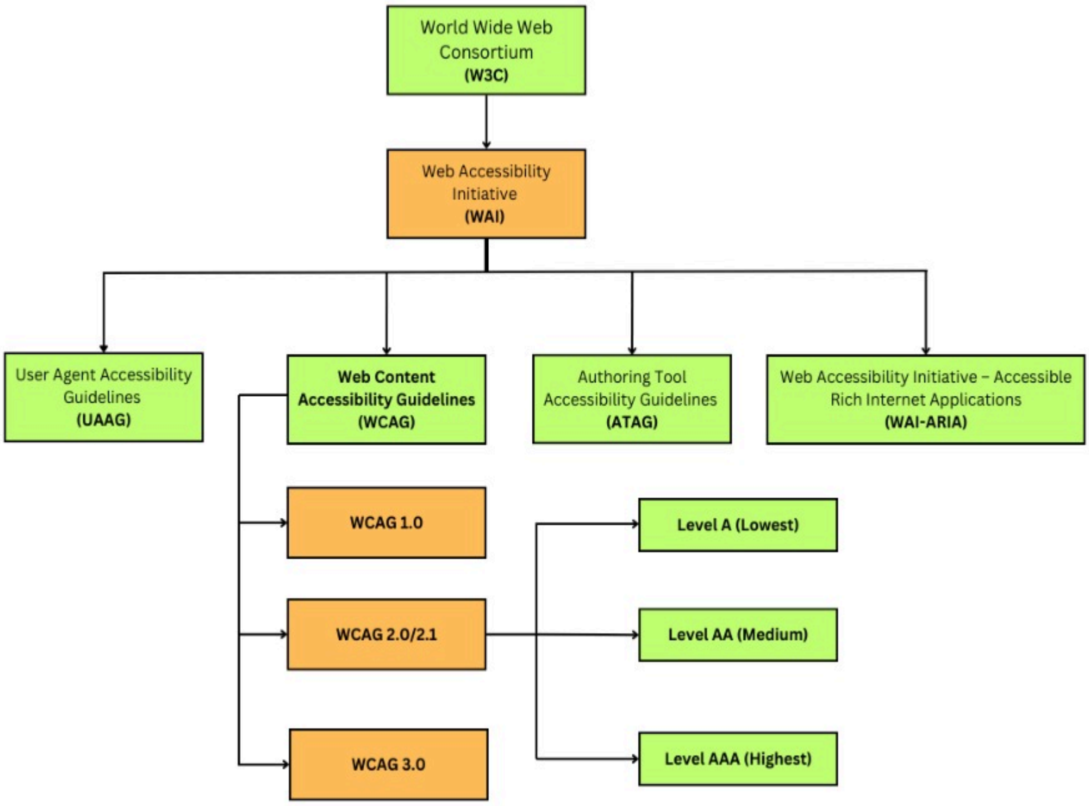
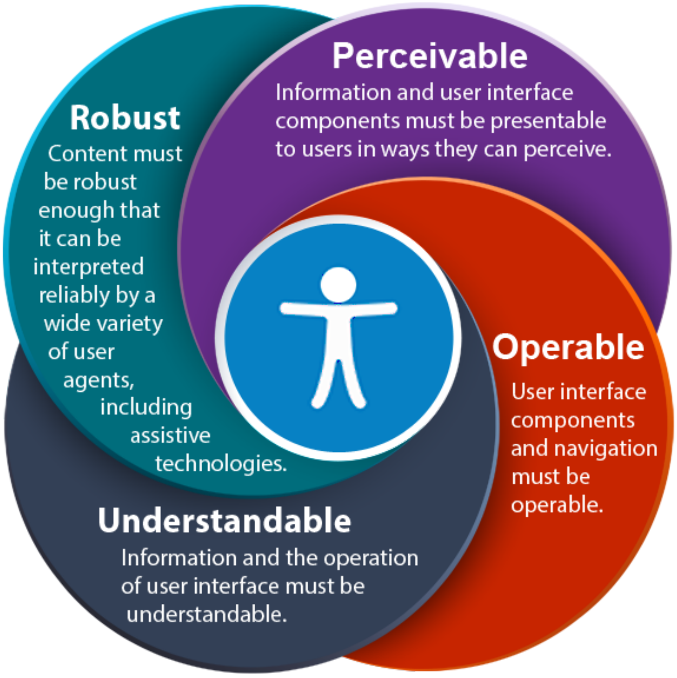

World Wide Web Consortium (W3C)
Web Accessibility Initiative (WAI)
User Agent Accessibility Guidelines (UAAG)
Web Content Accessibility Guidelines (WCAG)
Authoring Tool Accessibility Guidelines (ATAG)
Web Accessibility Initiative – Accessible Rich Internet Applications (WAI-ARIA)
WCAG 1.0
WCAG 2.0/2.1
WCAG 3.0
Level A (Lowest)
Level AA (Medium)
Level AAA (Highest)

Figure 1 showing Guidelines of Web accessibility



Perceivable
Information and user interface components must be presentable to users in ways they can perceive.
Robust
Content must be robust enough that it can be interpreted reliably by a wide variety of user agents, including assistive technologies.
Operable
User interface components and navigation must be operable.
Understandable
Information and the operation of user interface must be understandable.

Figure 2 showing the WCAG principles—Perceivable, Operable, Understandable, and Robust—are often referred to by the acronym "POUR."

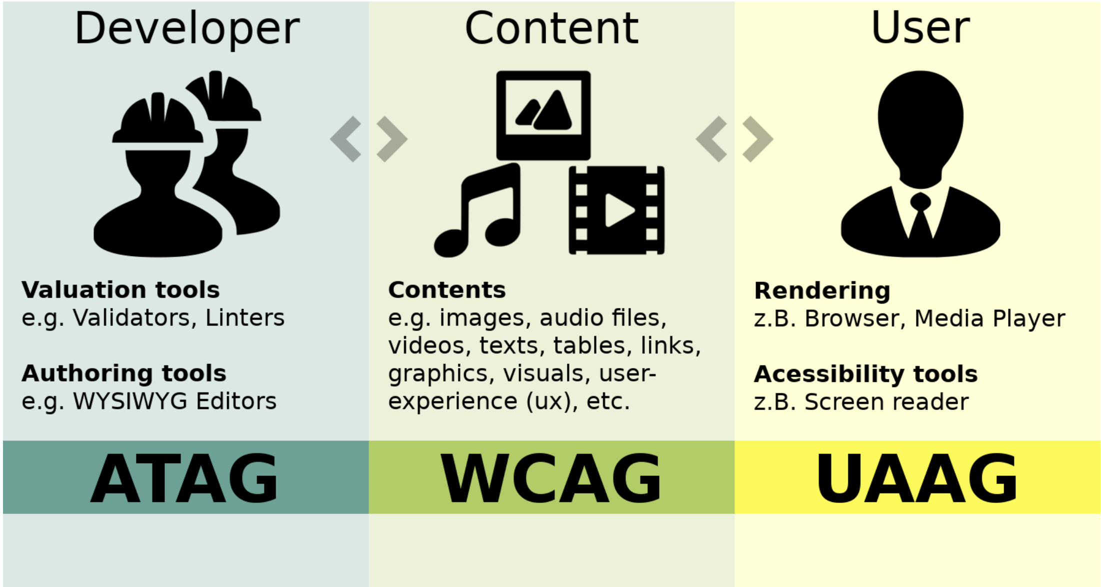


Figure 3 showing three web content accessibility guidelines

The International Organization for Standardization (ISO) has developed a number of guidelines addressing accessibility concerns pertaining to human-system interaction, with an emphasis on human-centered design and software, user interfaces, and PDF documents accessibility (such as ISO 9241, ISO Guide 71, ISO/IEC TR 29138, ISO/IEC 24751, and ISO 14289). Additionally, various other guidelines exist, including but not limited to Mobile Web Applications Best Practices (MWABP), Web Aim's Introduction to Web Accessibility, BBC Accessibility Guideline, Barrier Walkthrough Guide, IBM Accessibility, IMS Access for All, and GuAMA's Guide to the Development of Accessible Mobile Applications (Steinebach 2020).

C. CHALLENGES RELATED TO ACCESSIBILITY STANDARDS AND GUIDELINES

It has been observed that the proliferation of national laws and policies geared towards promoting accessibility of information and communication technologies (ICT), including the web, has led to a diversity of approaches in practice. Some of these laws and policies center on the recognition of the right to ICT as a human right, while others mandate that any ICT acquired by the government be accessible, and yet others stipulate that any ICT sold in a given market must be accessible. These are just a few of the approaches adopted by different jurisdictions. Still there are many developing countries which have not proposed any guidelines or laws regarding disabled people (Sloan and Horton 2014), for they are more emphasized on the equality, and the disability is described in different context, succession in accessibility and the usage of digital content, services, and products (Abuaddous, Jali et al. 2016).

D. BRIEF OVERVIEW OF DISABILITIES

Disabilities are a complex and diverse group of conditions that can affect a person's physical, mental, and emotional abilities. They can be present from birth, acquired later in life, or caused by an accident. Disabilities can range in severity from mild to profound and often impede an individual's capacity to perform routine activities of daily living (Lundqvist and Ström 2018). Web accessibility covers a broad range of disabilities that can impede access to web content, including but not limited to auditory, cognitive, neurological, physical, speech, and visual impairments. It is often exemplified by considering the needs of blind individuals, although this represents just a fraction of the population that could benefit from web accessibility measures, particularly among those with visual impairments. In reality, approximately one-fifth of the general population requires some form of web accessibility support, with people with cognitive disabilities being the largest group to benefit from it, as highlighted in a study conducted by Selovuo (Selovuo 2019, Jokinen 2020).

Below are some common disabilities:

1. VISUAL IMPAIRMENTS

People with visual impairments have a wide range of conditions that affect them differently. Some people with visual impairments may experience little to no disruption in their daily lives, while others may be completely blind. The internet can be a valuable tool for people with severe visual impairments to connect with the world around them. Even for those with milder impairments, web browsing may present minor inconveniences. However, by designing web applications with the specific needs of these users in mind, accessibility can be greatly enhanced, and the user experience improved (Lundqvist and Ström 2018).

2. AUDITORY DISABILITIES

Auditory disabilities encompass varying degrees of hearing loss in one or both ears, affecting individuals differently. Some individuals may experience difficulty in hearing speech, while others may have complete deafness. Several barriers exist for individuals with auditory disabilities, such as the absence of captions or transcripts for audio content, media players lacking caption display or customizable font options, and background noise interference. By implementing captions (text-based representations of audio content displayed alongside media, assisting those with hearing difficulties or deafness) and transcripts (written records of audio content that benefit individuals with hearing impairments and those who prefer reading), web developers can enhance website accessibility for individuals with auditory disabilities, ensuring equitable access to the internet's advantages. Moreover, additional recommendations for improving website accessibility for individuals with auditory disabilities include using clear and concise language to aid comprehension, avoiding jargon and technical terms to cater to a wider audience, providing transcripts for all audio content, and incorporating captions for all videos (Jokinen 2020)

3. MOTOR AND COGNITIVE DISABILITIES

Motion and motor disabilities encompass impairments that affect both complex movements, such as walking, and the limited mobility of specific body parts, such as the arms and hands. These limitations in mobility are not confined to older individuals but can also result from temporary disabilities caused by accidents, including sports-related injuries, impacting people across different age groups. Furthermore, individuals with permanent quadriplegia resulting from spinal cord injuries encounter difficulties in utilizing conventional input devices like mice or keyboards, thus relying on assistive technologies to access the web. Motor vehicle accidents are identified as the primary cause of spinal cord injuries. Friedman & Bryen suggested that individuals with cognitive disabilities commonly experience challenges related to fine motor control, hand-eye coordination, and finger dexterity (Friedman and Bryen 2007). Cognitive disabilities encompass a diverse range of impairments that impact various cognitive processes, such as learning, perception, concentration, and memory (Lundqvist and Ström 2018).

Figure 4 showing types of disabilities

E. IDENTIFIED USER REQUIREMENTS

The specified user requirements aim to address the diverse needs of individuals with disabilities, including visual impairments, cognitive impairments, motor impairments, and hearing impairments. Among the 40 user requirements, the majority, specifically 37, are focused on addressing the needs of individuals with visual impairments. Additionally, 31 requirements specifically target the elderly, while 25 requirements aim to accommodate individuals with cognitive impairments.

Furthermore, 22 requirements are designed to support people with disabilities in general, while 10 requirements are tailored for individuals with motor impairments. Lastly, 7 requirements are dedicated to enhancing accessibility for individuals with hearing impairments.Key user requirements include:

- Enabling large and adjustable font sizes to enhance readability for individuals with visual impairments.
- Offering carefully selected and adjustable color choices for font, background, and foreground to facilitate differentiation of page elements for people with visual impairments.
- Utilizing simplified language to enhance comprehension for individuals with cognitive impairments.
- Providing easy navigation options for individuals with motor impairments who may face challenges using a mouse or trackpad.
- Ensuring consistent and straightforward page layouts to aid individuals with cognitive impairments in understanding website structures.
- Supporting keyboard-based commands for website operation to accommodate users with motor impairments who rely on keyboard interaction.
- Maintaining high and adjustable contrast levels to facilitate element distinction for individuals with visual impairments.
- Offering sufficient and adjustable size and spacing of clickable and input elements to assist individuals with motor impairments in accurate selection.
- Incorporating closed captions, subtitles, and transcripts as alternative text for non-text content, such as audio and videos, to make content accessible to individuals with hearing impairments.
- Providing controls for speed, volume, pitch, playback, replay, stop, etc., to empower users with customization options.
- Avoiding information overload by adopting a simple structure and layout.
- Ensuring proper utilization of semantically meaningful HTML for improved screen reader comprehension.

- Supplying help documentation, tips, and guidance in audio or text format to assist users.
- Making visual content perceivable by other senses through techniques like audio descriptions or haptic feedback.
- Limiting open windows to mitigate confusion and frustration.
- Avoiding pop-up windows to enhance user experience.
- Effective management of focus to enable users to maintain awareness of their location on the page.
- Highlighting text when read out to aid users in following along.
- Employing headers and titles effectively to aid users in understanding the page structure.
- Clearly identifying links and their actions to provide users with clear expectations.
- Adjusting word, paragraph, and column spacing, length, width, and alignment to enhance readability for users.

By adhering to these guidelines, website developers can create accessible websites that cater to the needs of individuals with various disabilities. This inclusive approach ensures equal access to the internet's benefits for all users, regardless of their abilities (Steinebach 2020).

F. WEB ACCESSIBILITY: ASSISTIVE AND ADAPTIVE TECHNOLOGIES

Assistive technology serves as a comprehensive term that encompasses a diverse array of products and services aimed at aiding individuals in maintaining or enhancing their functional abilities and independence. This broad scope includes the utilization of assistive products like hearing aids and wheelchairs. Additionally, within the domain of assistive technology, there exists a specialized subcategory known as adaptive technology, specifically designed to address the unique requirements of individuals with disabilities. In contrast, assistive technology encompasses a wide range of solutions, whether commercially available or customized, with the common goal of improving the overall quality of life for its users (Jokinen 2020). Following are some different aspects of these technologies.

1. KEYBOARD NAVIGATION

Keyboard navigation enables users to navigate and interact with digital content using keyboard input alone, without relying on a mouse or other pointing device. It is particularly important for individuals with motor impairments or those who cannot use a traditional mouse. Keyboard navigation allows users to access and navigate websites, applications, and documents using keyboard shortcuts or tabbing between interactive elements.

2. CONTRAST AND TEXT PRESENTATION

Contrast refers to the difference in color and brightness between text and its background. It ensures legibility and readability for individuals with visual impairments or color vision deficiencies. Text presentation involves considerations such as font size, style, and spacing, which can impact readability and comprehension. Providing sufficient contrast and thoughtful text presentation enhances accessibility for all individuals W3C. (2018).

3. SCREEN READER COMPATIBILITY

Screen reader compatibility ensures that websites, applications, and documents are structured and designed in a way that allows screen readers to interpret and convey information audibly to individuals with visual impairments. It involves providing appropriate textual alternatives for non-text elements such as images, multimedia, and interactive elements, enabling screen readers to access and convey the content accurately.

4. PAGE LAYOUT AND TEXT CONTENT

Page layout refers to the arrangement and organization of content on a web page. A well-designed layout can enhance accessibility by providing clear and logical structure, facilitating ease of navigation and comprehension. Text content should be presented in a way that is concise, clear, and easy to understand, benefiting individuals with cognitive impairments or reading difficulties

G. WEB ACCESSIBILITY AROUND THE GLOBE

Around one billion individuals, comprising approximately 15% of the global population, live with disabilities that can impact their internet usage. Developing countries exhibit a higher prevalence of disabilities, with an estimated 110 million to 190 million people experiencing significant disabilities, accounting for roughly one-fifth of the global total. The number of individuals with disabilities is further influenced by the increasing global lifespan. Projections suggest that the aging population will more than double by 2050 and triple by 2100. According to a report from the Ministry of Internally Displaced Persons from the Occupied Territories, Labor, Health and Social Affairs of Georgia, the number of individuals receiving state social assistance related to disabilities increased from 118,651 in March 2015 to 127,132 in October 2020 (MoIDPOTLHSA, 2020). However, this figure represents only 3% of Georgia's total population, which falls below the more conservative estimates of global disability prevalence provided by the World Bank and the World Health Organization, indicating a prevalence of around 10%. Nevertheless, various sources indicate that similar patterns exist in Georgia. For instance, women with disabilities face heightened vulnerability and reduced access to government support or grants. Among internally displaced persons with disabilities, women and youth aged 15-24 face greater vulnerability compared to the general population of internally displaced persons and the average Georgian 0. The increasing prevalence of disabilities is a key factor underscoring the significance of website accessibility. As reported by the World Health Organization and the CDC, approximately 16% of the global population, and 26% of the population in the United States, are living with some kind of disability. This translates to over 1 billion individuals worldwide and around 86 million people in the U.S. who may encounter difficulties accessing websites that lack accessible design features. Based on data from Eurostat, it was found that 87% of individuals residing in the Euro area utilized the internet in 2019, showcasing a significant increase from the 62% reported in 2008. It has been observed that even countries with lower usage rates, such as Bulgaria, demonstrated growth, reaching a 68% usage rate in 2019. For individuals, especially those living in certain geographic or mental circumstances, the internet may serve as the sole means of accessing services and participating in societal activities. Web-based voting, for instance, has the potential to enhance participation and expand the reach of traditional voting methods.

However, a study conducted in Norway identified usability and accessibility issues within several prototype systems, indicating that the voting system itself can pose as a barrier despite the advantages of web-based voting (Fuglerud and Røssvoll 2012, Jokinen 2020). According to the Briefing Package for the creation of the Web Accessibility Initiative (WAI), there is a global population of over 750 million individuals with disabilities, many of whom are directly or indirectly impacted by the emergence of the Web. While the Internet has the potential to connect people across geographical boundaries, the presence of barriers within the web infrastructure poses a threat to the full participation of individuals with disabilities. Within the European Union (EU), people with disabilities constitute a significant portion of the population, yet they continue to face obstacles that hinder their complete integration into society. Findings from a 2012 Flash Eurobarometer survey confirmed the widely held belief that 93% of respondents acknowledged accessibility barriers as factors that make it more difficult for individuals with disabilities to access education, employment, voting, and the freedom to move around and go on holiday. The survey also revealed that 7 out of 10 Europeans recognized that improving accessibility in goods and services would have a substantial positive impact on the lives of people with disabilities. Additionally, 86% of Europeans agreed that establishing consistent accessibility solutions across Europe would facilitate travel, study, and work opportunities for individuals with disabilities within EU member states, thereby supporting the need for EU-wide legislation on accessibility (Ferri and Favalli 2018).

This study was designed to minimize the difficulties faced by individuals with disabilities. This research project aimed to develop a caption generation system that adheres to WCAG rules, thereby enabling print-impaired users, including the blind, partially sighted, and dyslexic individuals, to access websites effectively.

II. APPROCH AND SYSTEM DEVELOPMENT

A. LIBRARIES

Following libraries were imported for the set-up of the environment for the image captioning project.

- Numpy and pandas are used for data manipulation and analysis.
- Os is used to interact with the operating system.
- Tensorflow is the main deep learning framework used in this project.
- Tqdm is used to display progress bars during training.
- Imagedatagenerator, load_img, and img_to_array are used for image preprocessing.
- Tokenizer and pad_sequences are used for text preprocessing.
- Sequence and to_categorical are used for data preparation.
- Sequential, model, and various layers such as conv2d, maxpooling2d, lstm, dense, etc. Are used to build the deep learning models.
- Adam, modelcheckpoint, earlystopping, and reducelronplateau are used for model training and optimization.
- Warnings is used to suppress warning messages.
- Matplotlib and seaborn are used for data visualization.
- Textwrap is used to wrap long text descriptions.

B. READING AND VISUALIZATION OF IMAGES

The code is designed to retrieve and display images from two datasets, namely Flickr8k and Flickr30k. It begins by specifying the file paths for the image and caption files associated with both datasets. Next, the code utilizes the Pandas library to read the caption files and organize the data into two distinct data frames: flickr8k_data and flickr30k_data. These data frames serve as structured containers for storing the caption information obtained from the respective datasets.

1. Flickr8k

The Flickr8k dataset consists of a benchmark collection containing 8,000 images. Each image is accompanied by five distinct captions that offer detailed descriptions of the prominent entities and events depicted in the image. The dataset comprises images sourced from six diverse Flickr groups, encompassing a wide range of scenes and scenarios. It has been specifically curated for tasks related to sentence-based image description and search, making it a valuable resource for research and development in this domain.

2. Flickr30k

The Flickr30k dataset is a comprehensive collection comprising 31,783 images that serve as a benchmark for various computer vision tasks. Each image in the dataset is accompanied by five distinct captions, which offer detailed descriptions of the prominent entities and events depicted in the image. The images were sourced from Flickr and were deliberately selected to capture a wide range of scenes and situations, intentionally avoiding well-known individuals or locations. The dataset has been specifically designed to facilitate tasks related to sentence-based image description and search, making it a valuable resource for researchers and developers in the field of computer vision.

C. PRE-PROCESSING DATA

1. PRE-PROCESSING CAPTIONS TEXT

a. PRE-PROCESSING

Preprocessing includes:

- Conversion of sentences into lowercase
- Removal of special characters and numbers present in the text
- Removal of extra spaces
- Removal of single characters
- Addition of a starting and an ending tag to the sentences to indicate the beginning and the ending of a sentence

b. TOKENIZATION AND ENCODED REPRESENTATION

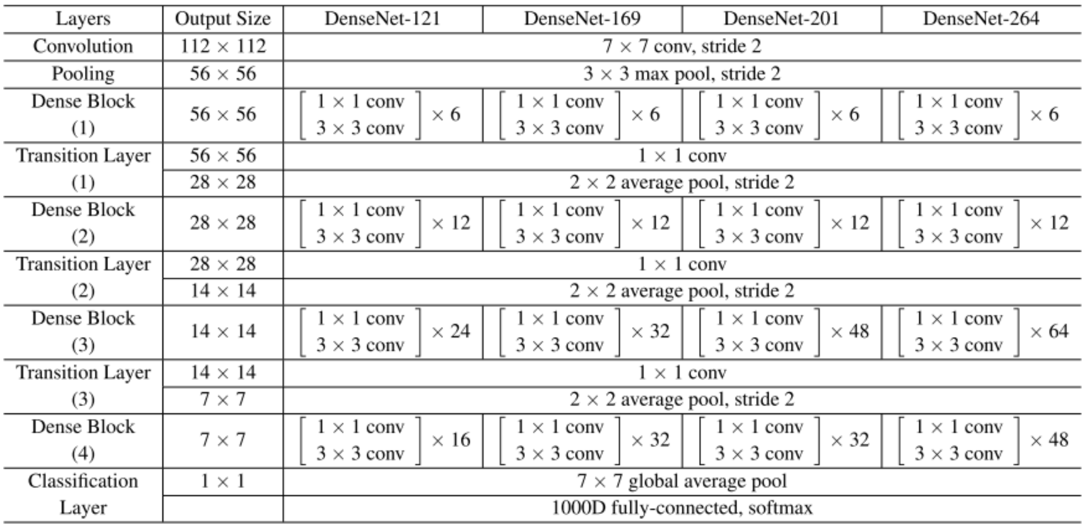
The words in a sentence are separated/tokenized and encoded in a one hot representation. These encodings are then passed to the embeddings layer to generate word embeddings.

c. IMAGE FEATURE EXTRACTION

The function "extract_image_features" takes as input a DataFrame containing image filenames, a directory path where the images are located, a pre-trained model, and a specified title for the saved file. Its main purpose is to extract features from the images using the provided model. Before extracting the features, the function checks if the features have already been extracted and saved in a file. If the features exist in a file, the function loads them and returns the loaded features. Otherwise, it proceeds to extract the features for each image using the pre-trained model. To extract the features, the function utilizes the "predict" method of the model. First, the image is preprocessed by loading it using "load_img" from the "keras.preprocessing.image" module. The preprocessed image is then converted into a numpy array using "img_to_array". Next, the array is normalized by dividing each element by 255 and expanded to include an additional dimension using "np.expand_dims". The features for each image are stored in a dictionary, with the image filename serving as the key.

d. DENSE NET

DenseNet is a convolutional neural network architecture that stands out for its unique layer connectivity pattern, allowing each layer to establish connections with all other layers in a forward direction. The architecture comprises multiple dense blocks, which consist of several convolutional layers and a concatenation layer that merges the outputs of all preceding layers within the block. To control the dimensionality of the feature maps, a transition layer is inserted after each dense block, reducing the dimensionality before passing the feature maps to the next dense block. The final output is generated by applying a global average pooling layer and a fully connected layer to the output of the last dense block. This sophisticated design empowers DenseNet to effectively capture and represent features by facilitating the flow of information and promoting the reuse of features across the network.



Layers	Output Size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	112 × 112	7 × 7 conv, stride 2			
Pooling	56 × 56	3 × 3 max pool, stride 2			
Dense Block (1)	56 × 56	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	56 × 56	1 × 1 conv			
	28 × 28	2 × 2 average pool, stride 2			
Dense Block (2)	28 × 28	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	28 × 28	1 × 1 conv			
	14 × 14	2 × 2 average pool, stride 2			
Dense Block (3)	14 × 14	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 24$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 48$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 64$
Transition Layer (3)	14 × 14	1 × 1 conv			
	7 × 7	2 × 2 average pool, stride 2			
Dense Block (4)	7 × 7	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 16$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 48$
Classification Layer	1 × 1	7 × 7 global average pool			
		1000D fully-connected, softmax			

Figure 5 showing the Dense Convolutional Network (Densnet) layers that connected with each other in feed-forward fashion.

e. CONNECTIVITY

DenseNet is characterized by its dense connectivity between layers, facilitating effective feature re-use and improving network performance and efficiency. In this architecture, every layer is connected to all other layers in a feed-forward manner. Within each dense block, the output of each layer is combined with the outputs of all preceding layers in the same block through concatenation. As a result, the resulting feature map contains a comprehensive integration of information from all previous layers, enabling thorough information aggregation and representation within the block.

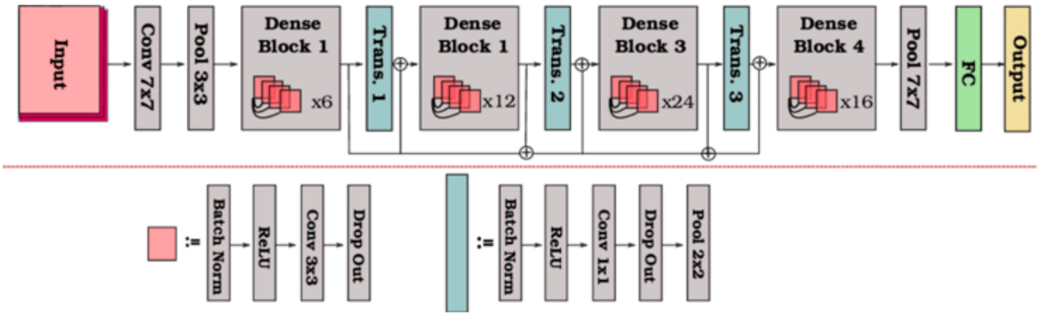
The i-th layer receives the feature-maps of all preceding layers, x0,…,Xl−1 as input:

$$X_l = H_l([X_0, X_1, \dots, X_{l-1}])$$

where Hl is the composite function that represents the l-th layer in the block, X0 is the input to the block, and Xl is the output of the block.

f.DENSEBLOCKS

In DenseNet, a DenseBlock represents a cohesive grouping of layers characterized by dense connectivity. Multiple layers are organized within each DenseBlock, where every layer establishes intricate connections with all other layers in a feed-forward manner. This arrangement facilitates the concatenation of the output from each layer with the outputs of all preceding layers in the block. As a result, the feature map generated by the DenseBlock encompasses a comprehensive aggregation of information obtained from all previous layers, enabling efficient integration and representation of information throughout the block.



G. Huang

Figure 6 showing the inside of denseblock and transition layers

g. GROWTH RATE

In DenseNet, the growth rate corresponds to the number of feature maps that are introduced to the network at each layer. Typically, a small value is chosen for the growth rate (such as 12 or 24) to effectively manage the number of parameters in the network and keep it at a low level. This strategic decision helps control the overall complexity of the model while still enabling the network to progressively capture and incorporate new features as it deepens.

h. BOTTLENECK LAYERS

Bottleneck layers are an integral part of DenseNet, serving to reduce the number of input feature maps prior to the convolutional layers. This strategic inclusion of bottleneck layers achieves various goals, such as minimizing the network's parameter count and enhancing its computational efficiency. The structure of a bottleneck layer encompasses a series of convolutional operations: a 1x1 convolutional layer that diminishes the number of input feature maps, followed by a 3x3 convolutional layer responsible for the primary convolution operation. Subsequently, another 1x1 convolutional layer expands the feature maps back to their original size.

To leverage the dense connectivity characteristic of DenseNet, the output of the bottleneck layer is concatenated with the outputs of all preceding layers within the block. This fusion of feature maps facilitates comprehensive information integration and representation. The resulting feature map undergoes further processing through an additional bottleneck layer and a final 1x1 convolutional layer, culminating in the generation of the block's output. By adopting this design, DenseNet adeptly manages information flow, optimizes parameter utilization, and fosters seamless feature integration across layers, thereby augmenting the network's overall performance and capacity for representation.

The formula for the output of a bottleneck layer is:

$$y = BN(ReLU(Conv_{1x1}\ (X)))$$
$$z = BN(ReLU(Conv_{3x3}\ (y)))$$
$$w = BN(Conv_{1x1}\ (z))$$

where X is the input feature map, Conv1x1 and Conv3x3 are 1x1 and 3x3 convolutional layers, respectively, BN is a batch normalization layer, and ReLU is the Rectified Linear Unit activation function.

The output of the bottleneck layer is the concatenation of the input feature map and the output of the final convolutional layer:

$$\text{out} = [X, w]$$

where X is the input feature map and w is the output of the final convolutional layer.

i.DENSENET121

DenseNet121 is a variant of the DenseNet architecture that stands out with its depth of 121 layers. This specific version consists of four dense blocks, where each block contains a different number of convolutional layers: 6, 12, 24, and 16, respectively. The growth rate parameter, set to 32, determines the number of feature maps added by the convolutional layers within each block. To address the dimensionality of the feature maps, DenseNet121 integrates transition layers. These layers employ a combination of a 1x1 convolutional operation and average pooling to effectively reduce the dimensionality of the feature maps. The input to DenseNet121 is an RGB image with dimensions of 224x224. This image serves as the initial data point for the network, initiating its computations and subsequent feature extraction processes.

j.DENSENET169

DenseNet169 is a variant of DenseNet that is known for its impressive depth, boasting a total of 169 layers. This particular configuration consists of four dense blocks, each containing a different number of convolutional layers: 6, 12, 32, and 32, respectively. With a growth rate of 32, the convolutional layers in each block contribute an equal number of feature maps. To manage the dimensionality of the feature maps, DenseNet169 incorporates transition layers. These layers combine 1x1 convolutions and average pooling to effectively reduce the dimensionality, facilitating more efficient information flow. The input to DenseNet169 is an RGB image with dimensions of 224x224 pixels. This image acts as the initial input for the network, initiating the computational processes and subsequent extraction of meaningful features.

k. DENSENET201

DenseNet201 stands as an advanced variant of DenseNet, characterized by an impressive depth of 201 layers. This particular iteration comprises four dense blocks, each with a unique composition of convolutional layers: 6, 12, 48, and 32, respectively. By setting the growth rate parameter to 32, the convolutional layers within each block contribute an equal number of feature maps, ensuring consistent information flow. To handle feature map dimensionality, DenseNet201 employs transition layers. These layers combine 1x1 convolutions with average pooling, effectively reducing the dimensionality of the feature maps. This reduction aids in enhancing computational efficiency during subsequent processing stages. When utilizing DenseNet201, the input expected by the network is an RGB image of size 224x224 pixels. This image serves as the initial input, initializing the network's computations and facilitating the extraction of valuable features from the image data.

l.RESNET

Convolutional neural network architecture ResNet is well known for its efficiency in deep network training. By introducing residual connections, it addresses the difficulties of deep network training. Each residual block in the ResNet network contains convolutional layers. The skip connection, which multiplies the block's input by its output, is the essential part of each residual block. With the help of this cutting-edge connection, the network can preserve crucial data from earlier layers and simplify gradient flow during training. The output of each residual block travels through a downsampling layer after passing through the convolutional layers and the skip connection. The spatial dimensionality of the feature maps is decreased by this layer, allowing for effective processing and the extraction of high-level characteristics. The following residual block in the network receives the reduced feature maps. Applying a global average pooling layer to the output of the last residual block yields the ResNet's final output. Following a fully linked layer for making predictions or conducting additional analysis, this layer compiles spatial data. ResNet can capture detailed representations, make use of leftover connections for deep network training, and perform well in a variety of computer vision applications because to this combination of processes.

m. RESIDUAL BLOCKS

A group of layers joined together in a special way to incorporate residual connections is known as a residual block. Multiple layers are stacked inside a residual block, and each LAYER is linked through a skip connection to the output of the layer that came before it. The block's input and output are combined via this skip connection using addition. By using this process, the network learns to recognize the residual mapping, which stands for the discrepancy between the block's input and output.

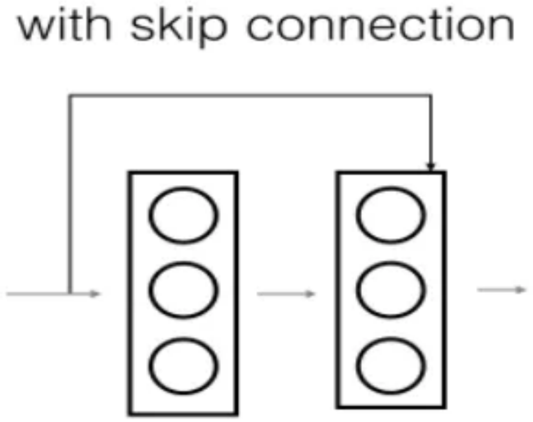


Figure 7 showing a residual block with a skip connection

For a residual block with a skip-connection from layer, l to l+2, the activation for layer l+2 can be computed as

$$a^{[l+2]} = g\left(z^{[l+2]} + a^{[l]}\right), \text{ where}$$
$$z^{[l+2]} = w^{[l+2]} * a^{[l+2]} + b^{[l+2]}$$

$$a^{[l+2]} = g\left(z^{[l+2]} + a^{[l]}\right), \text{ where}$$
$$z^{[l+2]} = w^{[l+2]} * a^{[l+2]} + b^{[l+2]}$$

Hence, equation 1 becomes,

$$a^{[l+2]} = g\left(w^{[l+2]} {}_{*} a^{[l+2]} + b^{[l+2]} + a^{[l]}\right)$$

n. DOWNSAMPLING

Downsampling aids in minimizing the number of parameters, hence enhancing its efficiency. It basically minimizes the feature maps' spatial dimentionality prior to passing the respective maps to the next. It is composed of three layers, i.e., 1x1 convolutional layer (reducing the input feature maps' number), 2x2 max pooling layer (applying max pooling operation), and 1x1 convolutional layer (bringing back the feature maps' numbers to their normal size. The result goes through another residual block and a 1x1 convolutional layer.

$$y = BN(\mathrm{ReLU}(\mathrm{Conv}_{1x1}\ (X)))$$
$$z = \mathrm{Max}_{\mathrm{Pool}_{2x2}\ (y)}\ (\mathrm{Conv}_{1x1}\ (z))$$

$$w = BN(\mathrm{Conv}_{1x1}\ (z))$$

where X is the input feature map, Conv1x1 is a 1x1 convolutional layer, MaxPool2x2 is a 2x2 max pooling layer, BN is a batch normalization layer, and ReLU is the Rectified Linear Unit activation function.

$$\text{out } = [X, w]$$

where X is the input feature map and w is the output of the final convolutional layer.

o. RESIDUAL NETWORKS – RESNET

Multiple residual blocks stack up to form residual network or deep ResNets which lead to the activation in the network when the activation of any particular layer turns zero earlier. If the activations for the layer l+2 tends to 0:

$a^{[l+2]}$ and $b^{[l+2]}$ tend to 0 and equation becomes,

$$a^{[l+2]} = g\left(a^{[l]}\right)$$

since with ReLU activation, g(a)= a for all a>0,

$$a^{[l+2]} = a^{[l]}$$

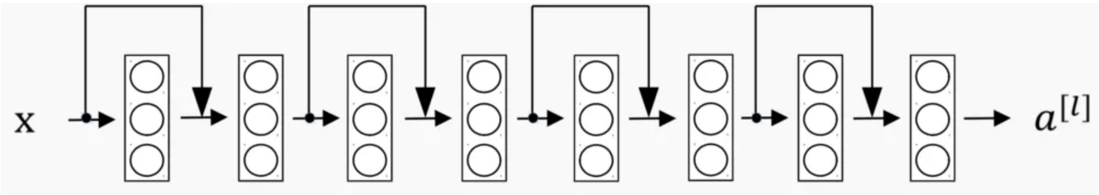


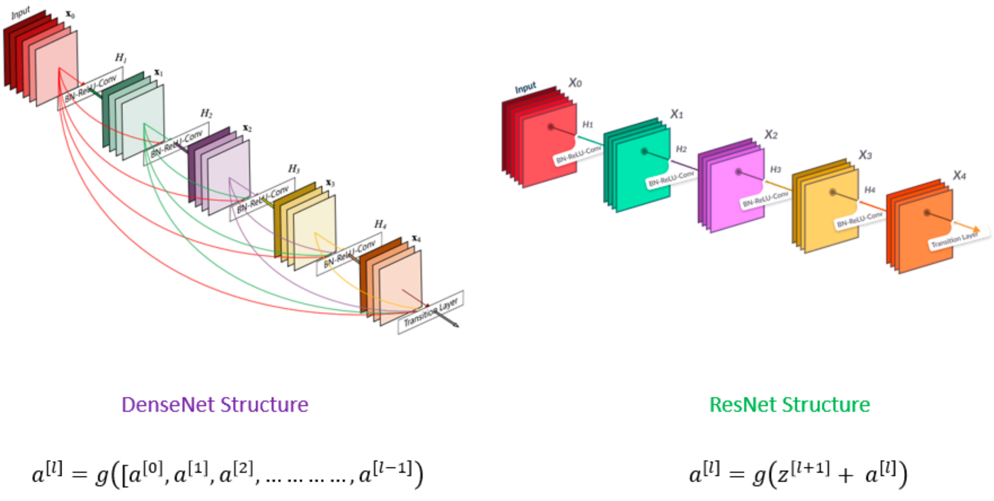
Figure 8 showing multiple residual blocks connected to form Residual network

p. RESNET 50

ResNet50, a variant of ResNet, comprises of 50 layers, four residual blocks. Each residual block has 3, 4, 6 and 3 convolutional layers, respectively. The spatial dimensionality is minimized by downsampling layers by using a stride of 2. In ResNet50, a 224x224 RGB image is used as an input to the network.

q. RESNET 101

ResNet101 comprises of 101 layers as its name suggests. It has the same number of residual blocks as of ResNet50, containing same convolutional layers with the exception of the third residual block which has 23 layers. Like ResNet50, a 224x224 RBG image is used, and the downsampling layers utilize same number of stride for spatial dimensionality.



DenseNet Structure

$$a^{[l]} = g([a^{[0]}, a^{[1]}, a^{[2]}, \dots \dots \dots \dots, a^{[l-1]})$$

ResNet Structure

$$a^{[l]} = g(z^{[l+1]} + a^{[l]})$$

Figure 9 showing DenseNet structure on left, while on right, a ResNet structure

r.DATA GENERATION

The data in Image Caption model is generated gradually in a batch manner as per requirement because it is a high resource utilizing procedure. The data can not be loaded all at the same time in the memory. The training model needs to be loaded with image embeddings and caption text embeddings. While inferring the output, the text embeddings of the resulting caption are proofread and passed word by word.

III. MODELLING

A. ATTENTION MODEL

The provided code represents a deep learning algorithm that employs the attention mechanism for image captioning. This algorithm takes both images and a sequence of words as inputs and aims to generate a descriptive caption for a given image. The implementation of this algorithm was carried out using the Keras library with the Tensorflow backend. The model design consists of two input layers: one for the image and another for the sequence of words. The image input undergoes processing through a dense layer with 256 units, utilizing the ReLU activation function. The resulting output from this layer is reshaped to (1, 256) and combined with the output of the attention layer.

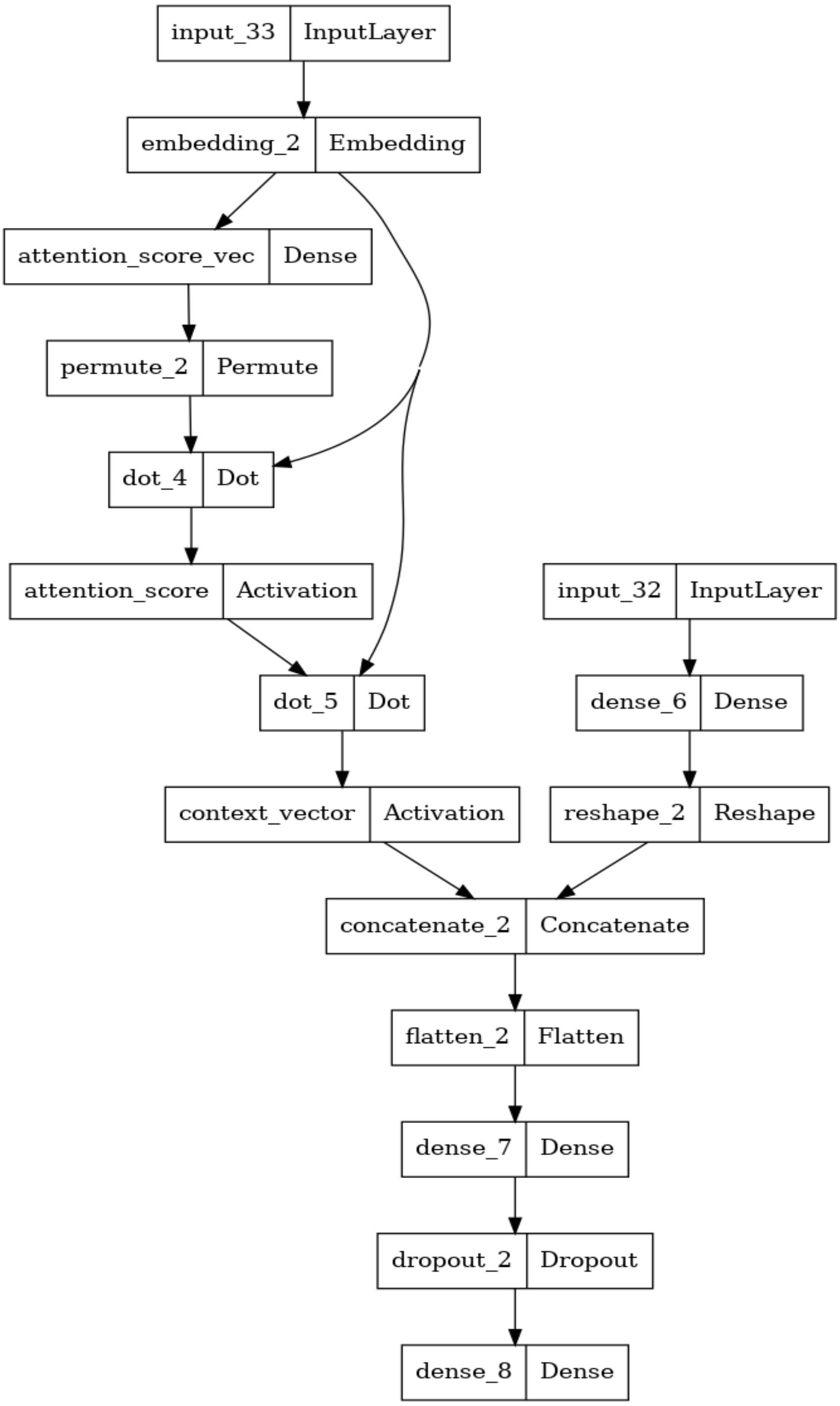


Figure 10 showing Attention model

B. LSTM + CNN MODEL

The code presents the implementation of the "lstm_model" function, which generates a deep learning model for image captioning through LSTM utilization. The function executes the sequential steps. The function requires three parameters: embedding_size, max_length, and vocab_size. Two input layers are generated in the process: input1 which deals with the image attributes, and input2 which handles the text features. The image characteristics undergo processing via a dense layer consisting of 256 units and activated with ReLU, adopting a formal language and terms appropriate for an expert audience seeking general information. The resulting layer of high computational density is reformulated to possess a configuration of (1, 256). The characteristics of the text undergo a process of transmission via an embedding layer consisting of 256 units, without any form of masking. The concatenation of the reshaped image features and embedded text features occurs along the first axis in a structured manner. The fused attributes undergo processing via an LSTM layer equipped with 256 units. The LSTM layer's output is transmitted through a dropout layer that has a rate of 0.5, as it is required in the given model. The summation of the output from the dropout layer and the image features takes place. The summation is processed via a dense layer that contains 128 units and utilizes ReLU activation, in adherence to formal language and expert audience. The output of the dense layer is passed through another dropout layer with a rate of 0.5. The ultimate result is generated via a compressed layer consisting of a vocabulary quota of units and activated by softmax. The model is constructed using the categorical cross-entropy loss function, the Adam optimizer, and the accuracy metric, all of which are fundamental components of its architecture and design. The compiled model is what gets returned by the function.

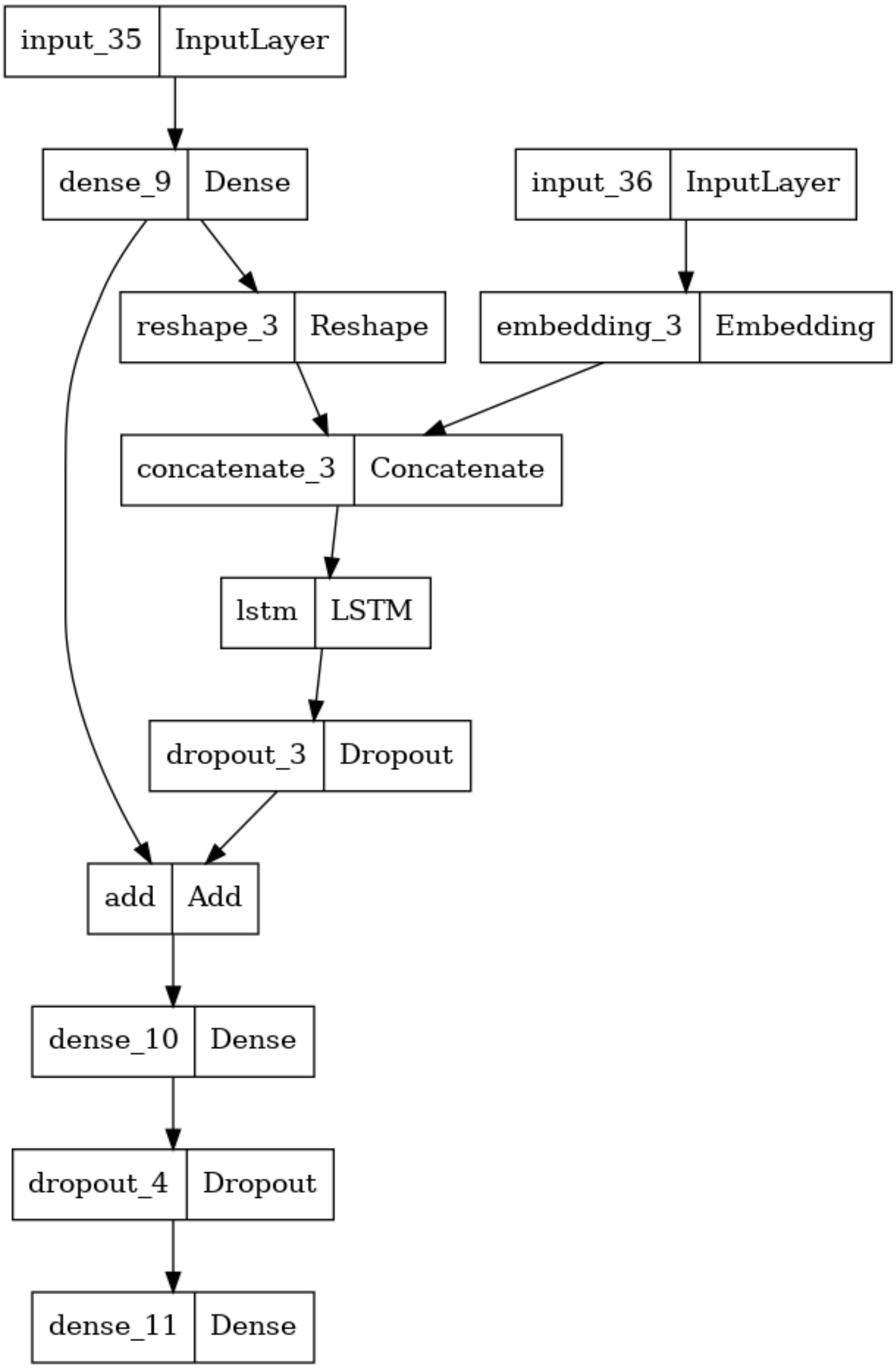


Figure 11 showing LSTM+CNN model

IV. EXPERIMENTAL EVALUATION

A. MODEL EVALUATION

The provided code defines a function called "train" that is responsible for training a deep learning model for image captioning by processing the dataset, preparing the data generators, training the model, and returning relevant information for further analysis and evaluation. The function takes four parameters: dataset_type, model_title, num_epochs, and caption_model, and within the function, the code performs the following steps, i.e., it extracts the key features from the images and tokenizes the captions, custom data generators are created for the training and validation sets, the function sets up callbacks for early stopping and learning rate reduction, the model is trained on the training data, using the specified number of epochs, after training, the function returns a dictionary containing various information, including the trained model, training history, tokenizer, training and test sets, maximum caption length, and vocabulary size.

The main objective of this code is to facilitate the training and storage of multiple image captioning models by systematically exploring different combinations of datasets, pre-trained image models, architectures, and epochs. The code starts by defining lists that contain the datasets, models, architectures, and epochs to be used. By utilizing the `itertools.product` function, the code iterates through all possible combinations of these elements. The code examines each combination and verifies if a saved model file exists. If a saved model is present, it is loaded and included in the results dictionary. However, if a saved model file is not found, the code proceeds to train a new model using the `train` function. The resulting dictionary, referred to as `result_dict`, is then saved to a specified file. Furthermore, the `result_dict` is incorporated into the overall results dictionary. Throughout the execution of the code, progress messages are displayed to keep the user informed about the ongoing training and loading of models. Once the code completes its execution, the final results dictionary contains essential information such as the trained models, training history, tokenizer, training and testing data, maximum caption length, and vocabulary size. Each combination of dataset, model, architecture, and number of epochs is represented in the results, enabling comprehensive analysis and evaluation of the image captioning models.

B. LEARNING CURVE

The 'loss' and 'val_loss' keys for the training and validation loss values, respectively, are assumed to be present in the history object by this function. When comparing the validation loss of multiple models, this function is helpful. Models that do not meet a particular set of criteria can be eliminated using the condition function.

C. LEARNING CURVE BY DATASET (8K OR 30)

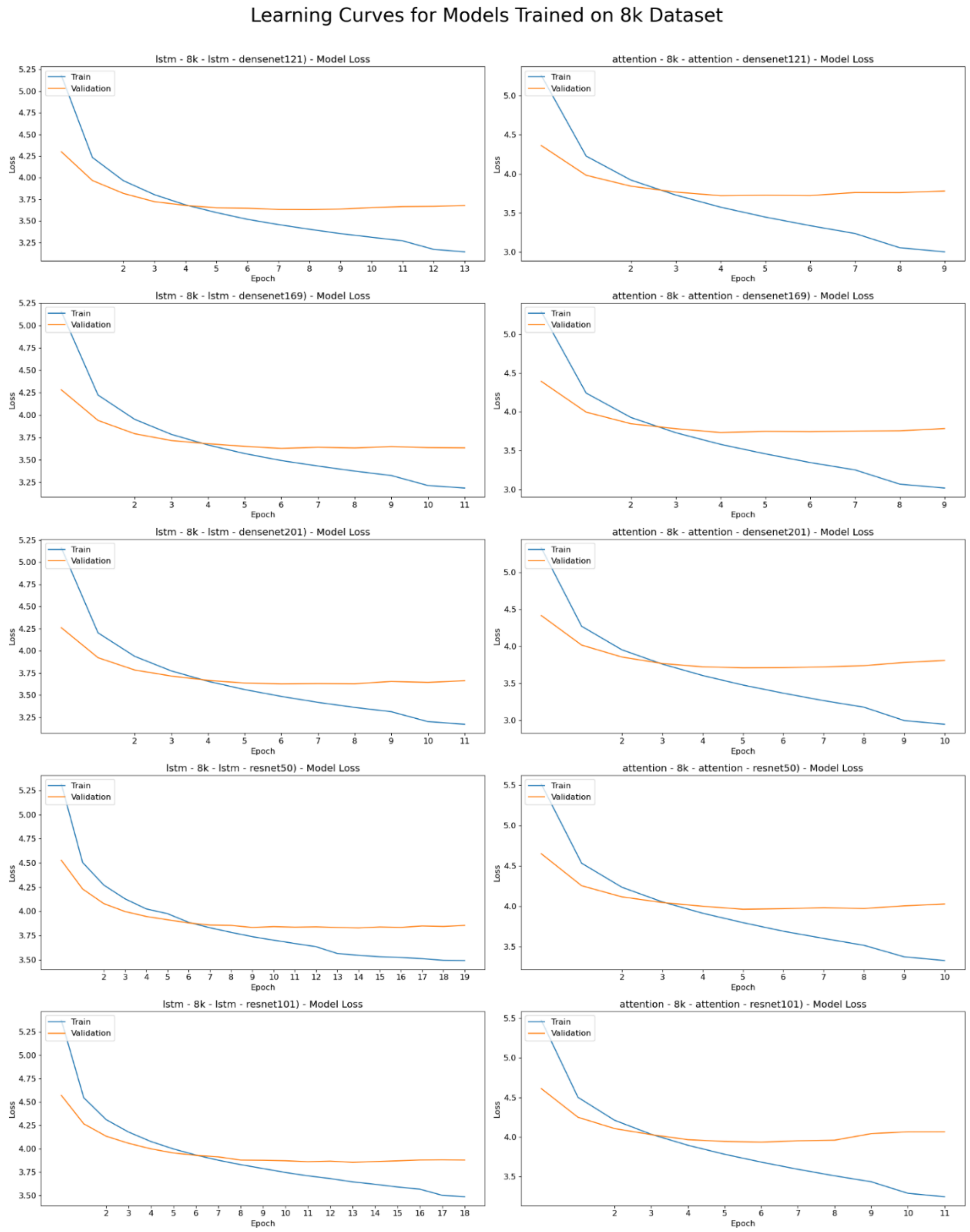


Figure 12 showing Learning curves for models trained on 8k dataset

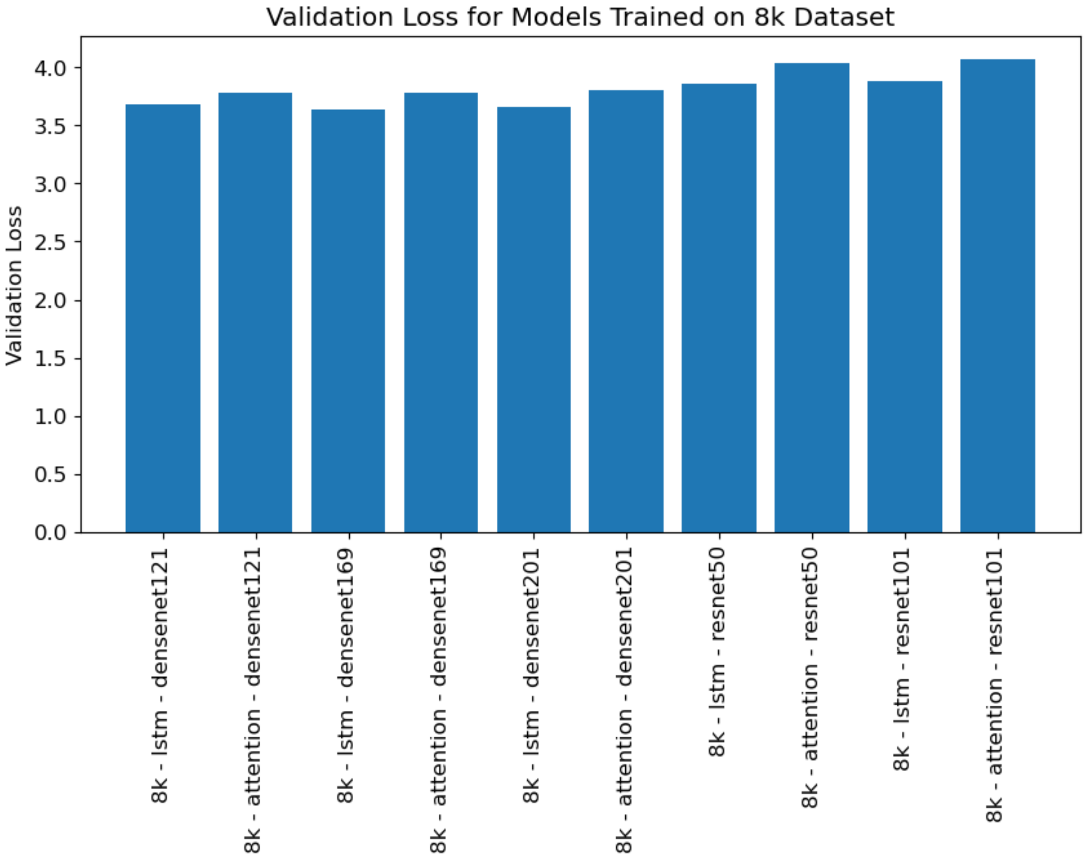


Figure 13 showing Loss for models trained on 8k dataset

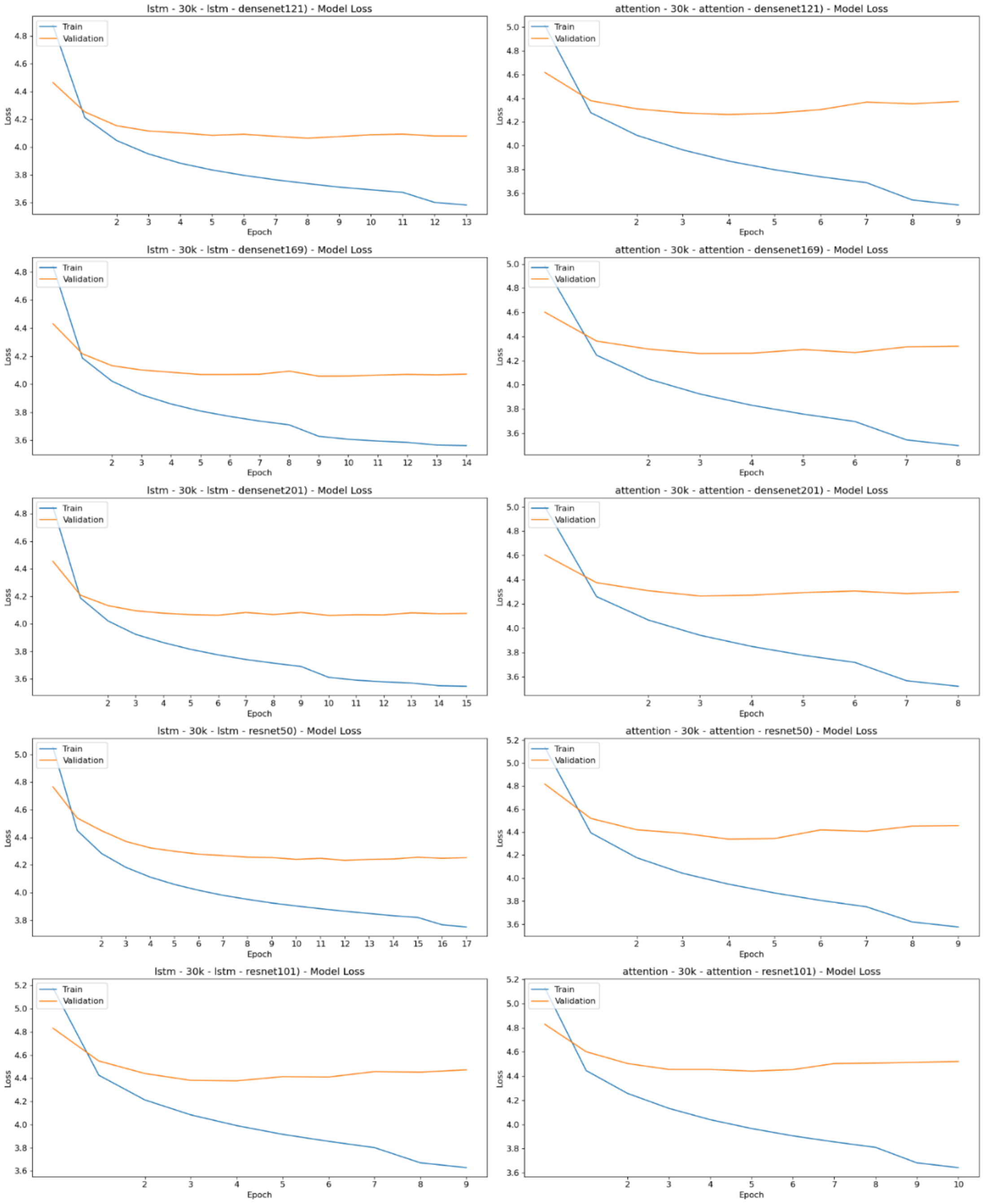


Figure 14 showing Learning curves for models trained on 30k dataset

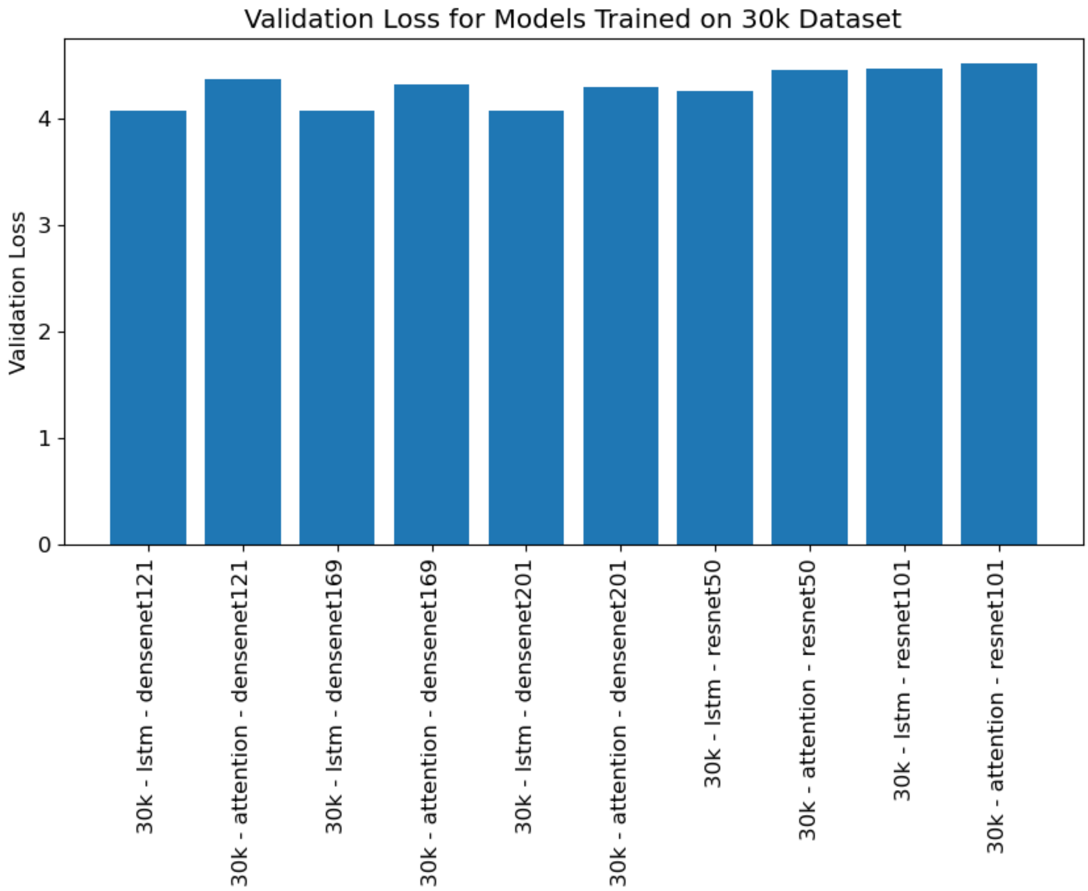
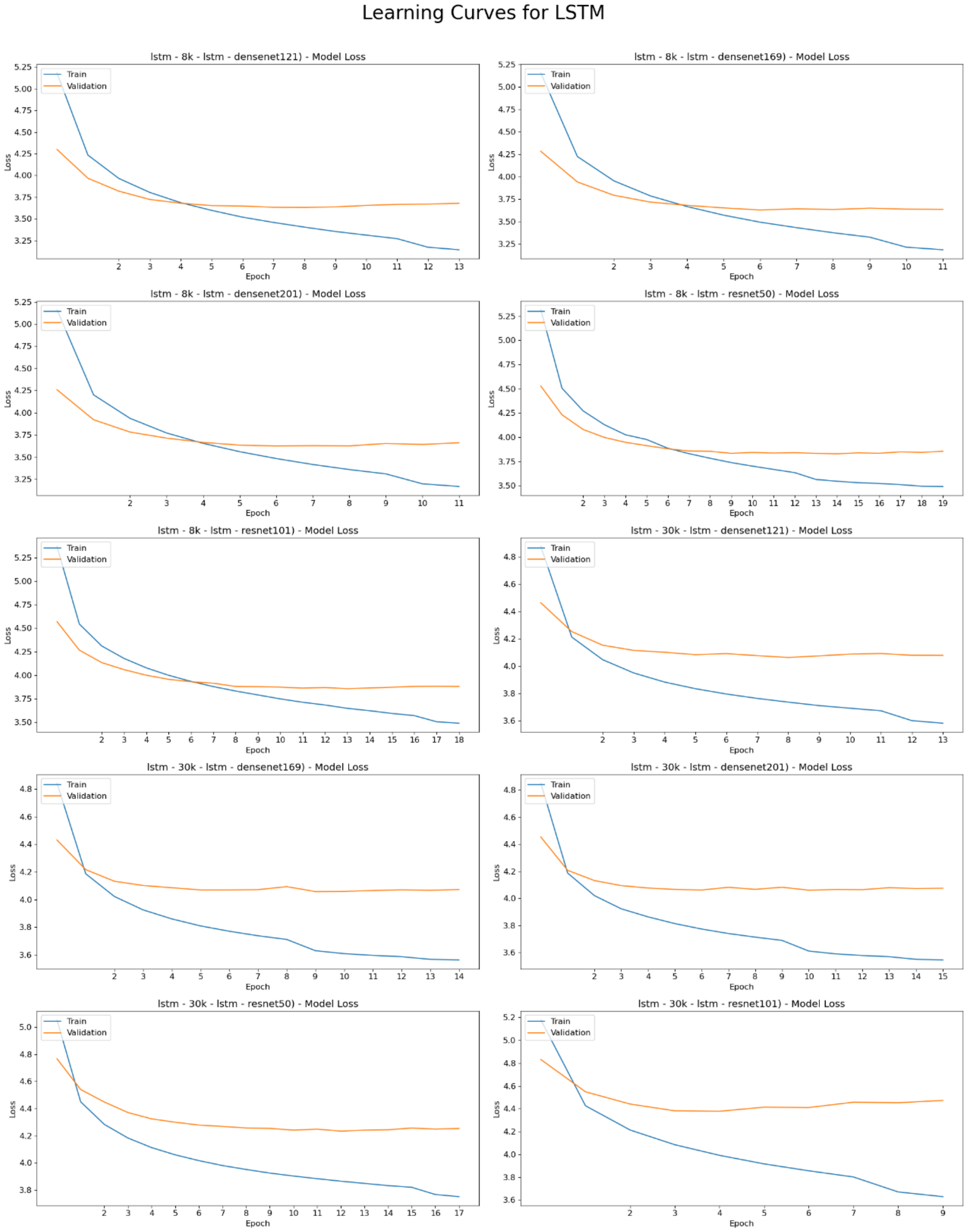


Figure 15 showing Loss for models trained on 30k dataset

D. LEARNING CURVE BY ARCHITECTURE



Learning Curves for LSTM

lstm - 8k - lstm - densenet121) - Model Loss

lstm - 8k - lstm - densenet169) - Model Loss

lstm - 8k - lstm - densenet201) - Model Loss

lstm - 8k - lstm - resnet50) - Model Loss

lstm - 8k - lstm - resnet101) - Model Loss

lstm - 30k - lstm - densenet121) - Model Loss

lstm - 30k - lstm - densenet169) - Model Loss

lstm - 30k - lstm - densenet201) - Model Loss

lstm - 30k - lstm - resnet50) - Model Loss

lstm - 30k - lstm - resnet101) - Model Loss

Figure 16 showing Learning curves for LSTM

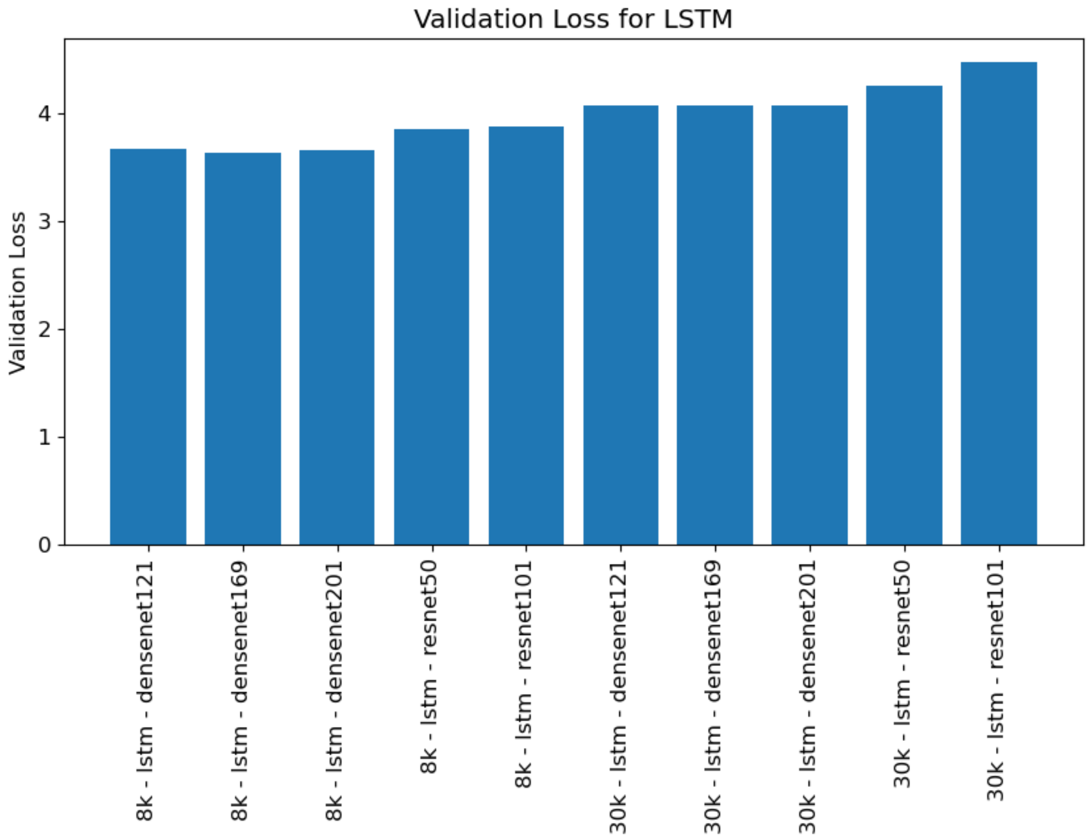
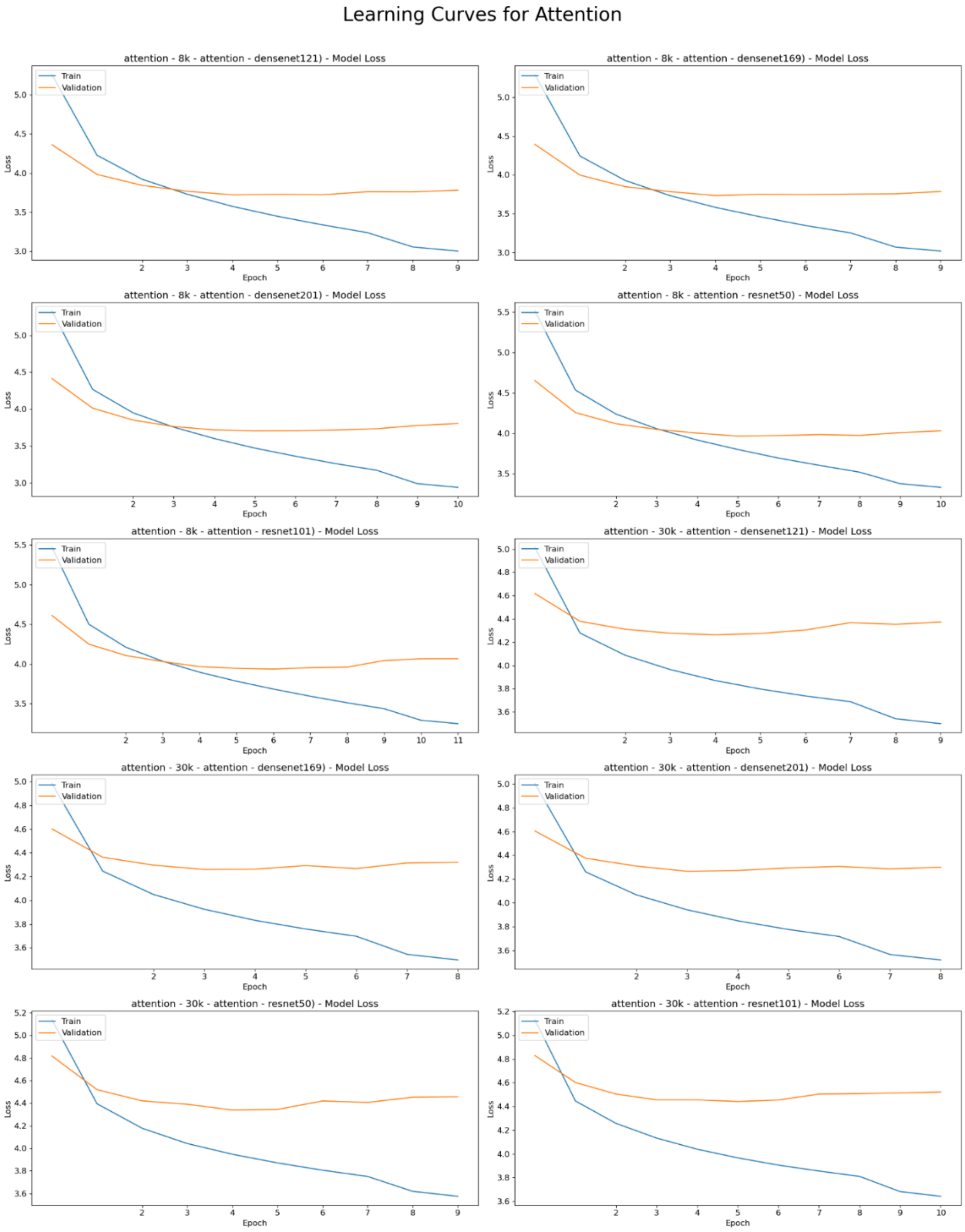


Figure 17 showing validation loss for LSTM



Learning Curves for Attention

attention - 8k - attention - densenet121) - Model Loss

attention - 8k - attention - densenet169) - Model Loss

attention - 8k - attention - densenet201) - Model Loss

attention - 8k - attention - resnet50) - Model Loss

attention - 8k - attention - resnet101) - Model Loss

attention - 30k - attention - densenet121) - Model Loss

attention - 30k - attention - densenet169) - Model Loss

attention - 30k - attention - densenet201) - Model Loss

attention - 30k - attention - resnet50) - Model Loss

attention - 30k - attention - resnet101) - Model Loss

Figure 18 showing Learning curves for Attention

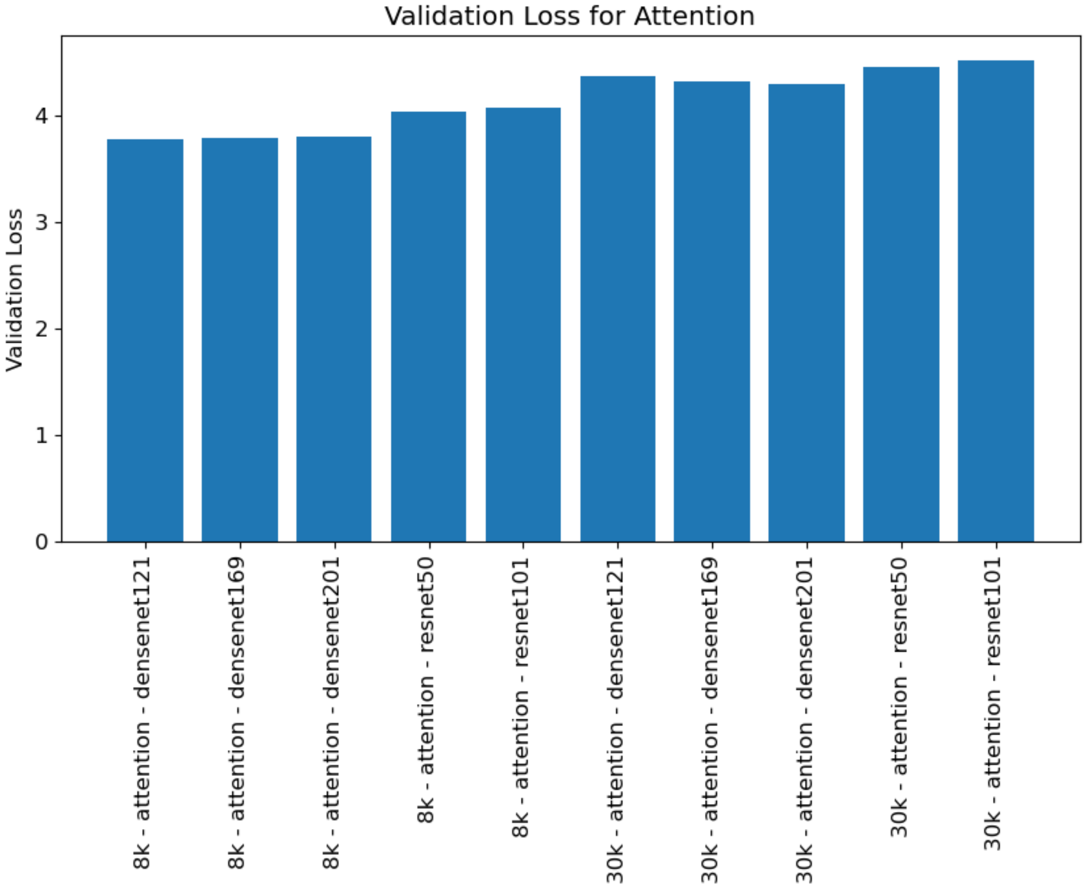
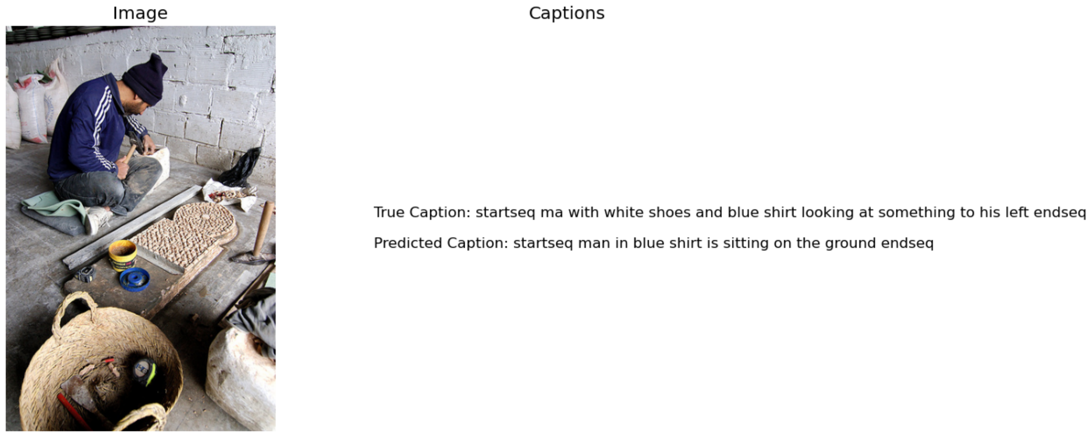


Figure 19 showing validation loss for Attention

E. TESTING MODEL

For each type of model and dataset, this function first obtains the pre-trained model and data paths. The pre-trained model is then used to extract image features for the given dataset. At long last, it applies the removed highlights to the info image_data and returns a rundown of the subsequent picture highlights.

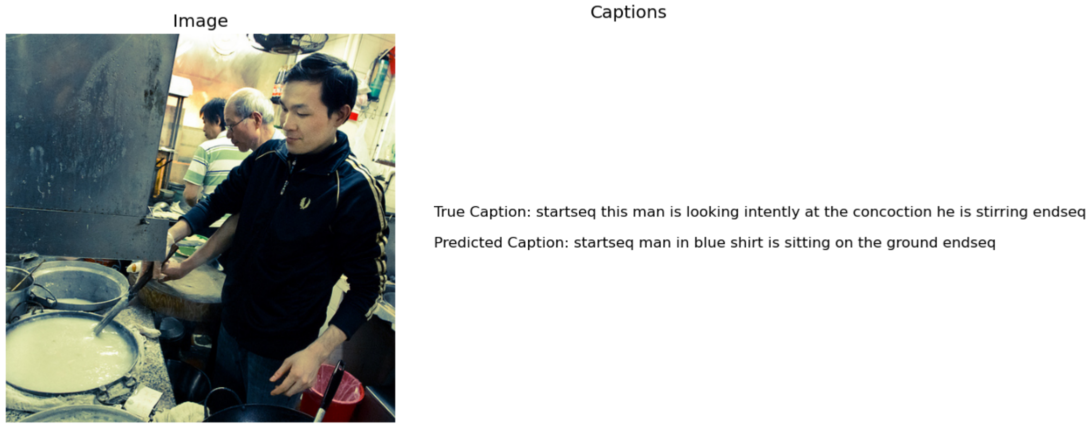
F. PREDICTED RESULTS



True Caption: startseq ma with white shoes and blue shirt looking at something to his left endseq

Predicted Caption: startseq man in blue shirt is sitting on the ground endseq

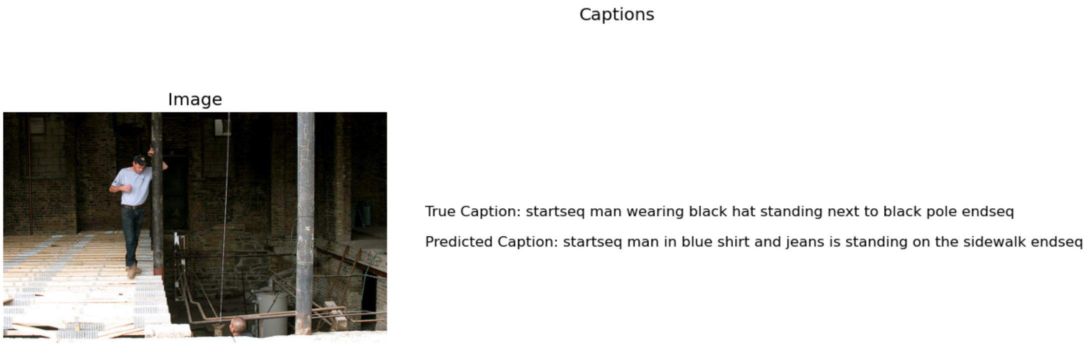
Figure 20 showing the predicted caption which we got from this research along with true caption (Sample Result 1).



True Caption: startseq this man is looking intently at the concoction he is stirring endseq

Predicted Caption: startseq man in blue shirt is sitting on the ground endseq

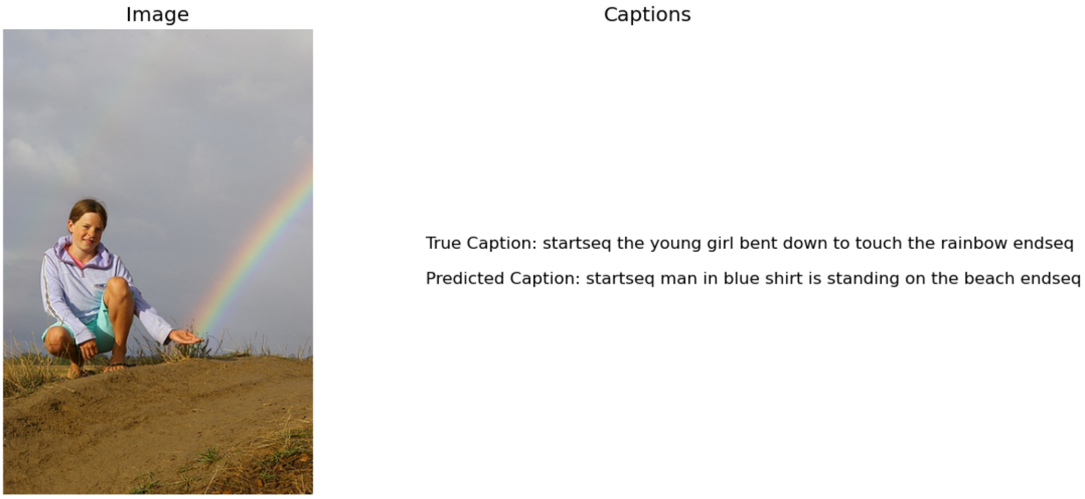
Figure 21 showing the predicted caption which we got from this research along with true caption (Sample Result 2).



True Caption: startseq man wearing black hat standing next to black pole endseq

Predicted Caption: startseq man in blue shirt and jeans is standing on the sidewalk endseq

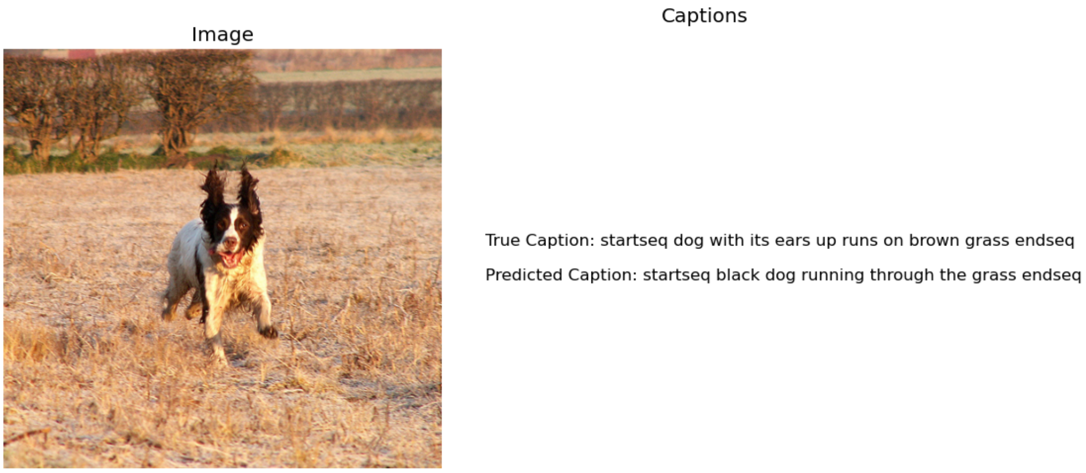
Figure 22 showing the predicted caption which we got from this research along with true caption (Sample Result 3).



True Caption: startseq the young girl bent down to touch the rainbow endseq

Predicted Caption: startseq man in blue shirt is standing on the beach endseq

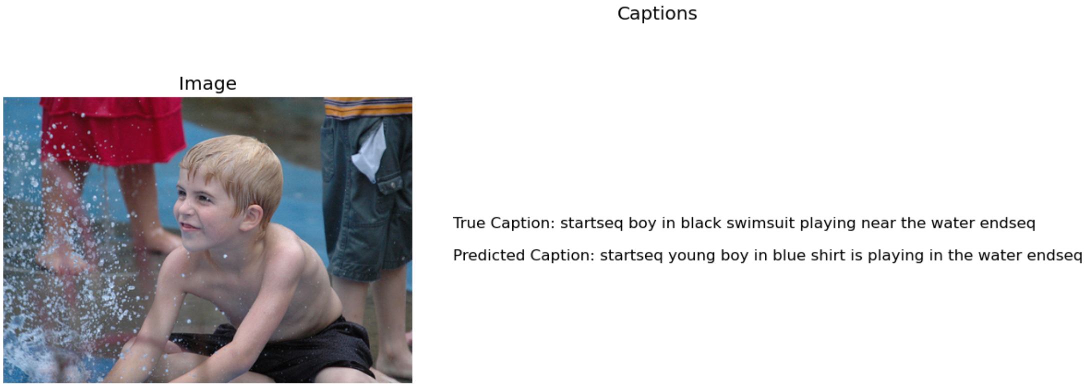
Figure 23 showing the predicted caption which we got from this research along with true caption (Sample Result 4).



True Caption: startseq dog with its ears up runs on brown grass endseq

Predicted Caption: startseq black dog running through the grass endseq

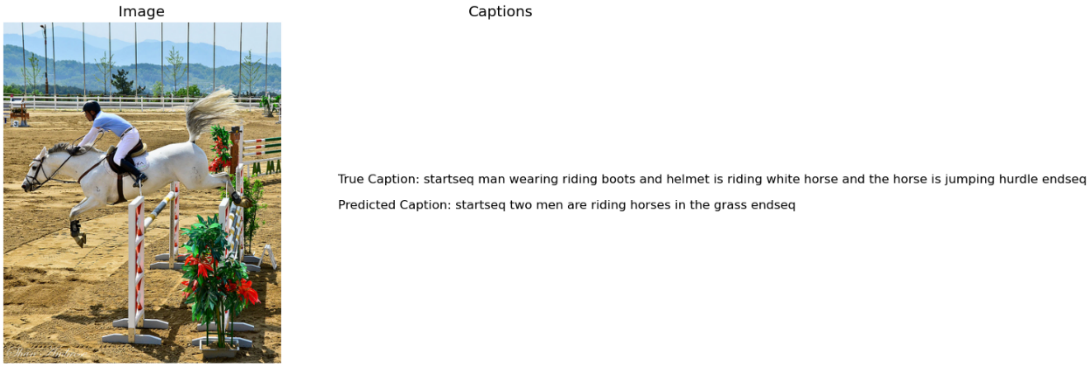
Figure 24 showing the predicted caption which we got from this research along with true caption (Sample Result 5).



True Caption: startseq boy in black swimsuit playing near the water endseq

Predicted Caption: startseq young boy in blue shirt is playing in the water endseq

Figure 25 showing the predicted caption which we got from this research along with true caption (Sample Result 6).



True Caption: startseq man wearing riding boots and helmet is riding white horse and the horse is jumping hurdle endseq

Predicted Caption: startseq two men are riding horses in the grass endseq

Figure 26 showing the predicted caption which we got from this research along with true caption (Sample Result 7).

The above figures (20-26) are the predicted results with their captions generated by the pre-trained models.

G. PERFORMANCE COMPARISON

Consider a scenario where you have a friend who is in the process of learning how to draw. Occasionally, their drawings resemble what they intended to depict, while other times they fall short of their desired outcome. Now, envision yourself attempting to assist your friend in improving their drawing skills. You could simply glance at their artwork and provide generic feedback such as "good job" or "not good job," but that wouldn't be particularly helpful. Instead, you might opt to offer more specific feedback, such as saying, "I can see that you were trying to draw a cat, but it appears more like a dog." Similarly, when we utilize a computer to describe an image, we strive for maximum accuracy in the generated description. However, similar to your friend's drawings, the computer's descriptions may sometimes be close to the mark but not entirely accurate. Rather than providing generic feedback like "good job" or "not good job," we employ evaluation metrics such as BLEU (Bilingual Evaluation Understudy) or WER (Word Error Rate) to assess the precision of the computer-generated descriptions. These metrics act as specialized tools that enable us to measure the proximity of the computer's description to the actual image. For instance, if the computer is tasked with describing an image of a cat and produces the description "a small furry animal with pointy ears and a long tail," it would be considered reasonably accurate. However, if it generates the description "a big green monster with wings and horns," that would clearly deviate from the intended depiction. By utilizing BLEU or WER, we can quantify the proximity of the computer's description to the actual image. This information can then be leveraged to improve the computer's ability to generate more accurate descriptions in the future.

H. PERFORMANCE MEASURES

1. BLEU

BLEU (bilingual evaluation understudy) is an evaluation metric employed to assess the quality of machine-translated text by comparing it to one or more reference translations. It measures the similarity between the machine-generated text and the reference translations based on the overlap of n-grams, which are contiguous sequences of n words. The BLEU score is computed using a formula that takes into account several factors. One of these factors is the brevity penalty (BP), which adjusts the score based on the length of the machine-generated text in relation to the reference translations. Additionally, the score considers the precision of each n-gram, which is the ratio of the number of times an n-gram appears in the machine-generated text and the number of total n-grams in the machine-generated text. The BLEU score ranges from 0 to 1, with 1 indicating a perfect match between the machine-generated text and the reference translations. BLEU is widely utilized in various natural language processing tasks, including machine translation and text summarization. It provides a quantitative measure to evaluate the quality of generated text and allows researchers and practitioners to compare different models and approaches in these domains.

$$BLEU = BP \cdot \exp\left(\frac{1}{n}\sum_{i=1}^{n} w_i \log{(p_i)}\right)$$

Were,

w_i is the weight assigned to the i-th n-gram and p_i is the precision of the i-th n-gram.

2. WER

Word error rate (WER) is a metric employed to assess the performance of speech recognition or machine translation systems by quantifying the disparity between the machine-generated text and the reference text in terms of word errors. It is calculated as the ratio of the total number of word errors (substitutions, deletions, and insertions) to the total number of words in the reference text. The WER score is determined using a formula that considers the number of substitutions (words in the machine-generated text that differ from the reference text), deletions (words missing from the machine-generated text compared to the reference text), insertions (words present in the machine-generated text but not in the reference text), and the total number of words in the reference text. WER serves as a widely adopted metric in speech recognition and machine translation tasks, enabling the evaluation of the accuracy and performance of generated text. It provides a quantitative measure to compare different systems or track the progress of a system over time, aiding in the advancement and refinement of these technologies.

$$WER = \frac{S + D + I}{N}$$

Where,

S is the number of substitutions, D is the number of deletions, I is the number of insertions and N is the total number of words in the reference text.

I. CALCULATION

This programme computes evaluation metrics for various dataset, model, architecture, and epoch number combinations. It creates an empty dataframe to record the findings and loops over all possible combinations of the parameters listed above. It obtains the relevant information from a dictionary of outcomes for each combination, including the tokenizer, test set, maximum sequence length, and trained model. The trained model is then used to create predicted captions for a selection of test pictures, and the BLEU and WER scores for each prediction are calculated in comparison to the genuine caption. The average BLEU and WER scores are stored to the dataframe that was previously initialized. During the loop, the code displays a progress bar using the tqdm library.

J. COMPARE BY DATASET AND ARCHITECTURE

Plots the specified metric (either BLEU or WER) for each model in both datasets for the specified architecture.

1. WER BY MODEL (LSTM)

Table 1 LSTM model evaluation by using WER on different Densenet layers

Model	30K	8K
densenet121	0.832318	0.779383
densenet169	0.866481	0.825615
densenet201	0.834433	0.83237
resnet101	0.914629	0.898758
resnet50	0.901348	0.84575

Discussing Table 2, Densenet121 generally had the lowest WER scores when comparing models with the LSTM architecture on various datasets, indicating greater caption generation accuracy. The resnet101 model had the highest WER scores among the resnet models, indicating lower accuracy than the other models. The size of the dataset also played a role. Models trained on the 30K dataset typically had slightly higher WER scores than models trained on the 8K dataset.

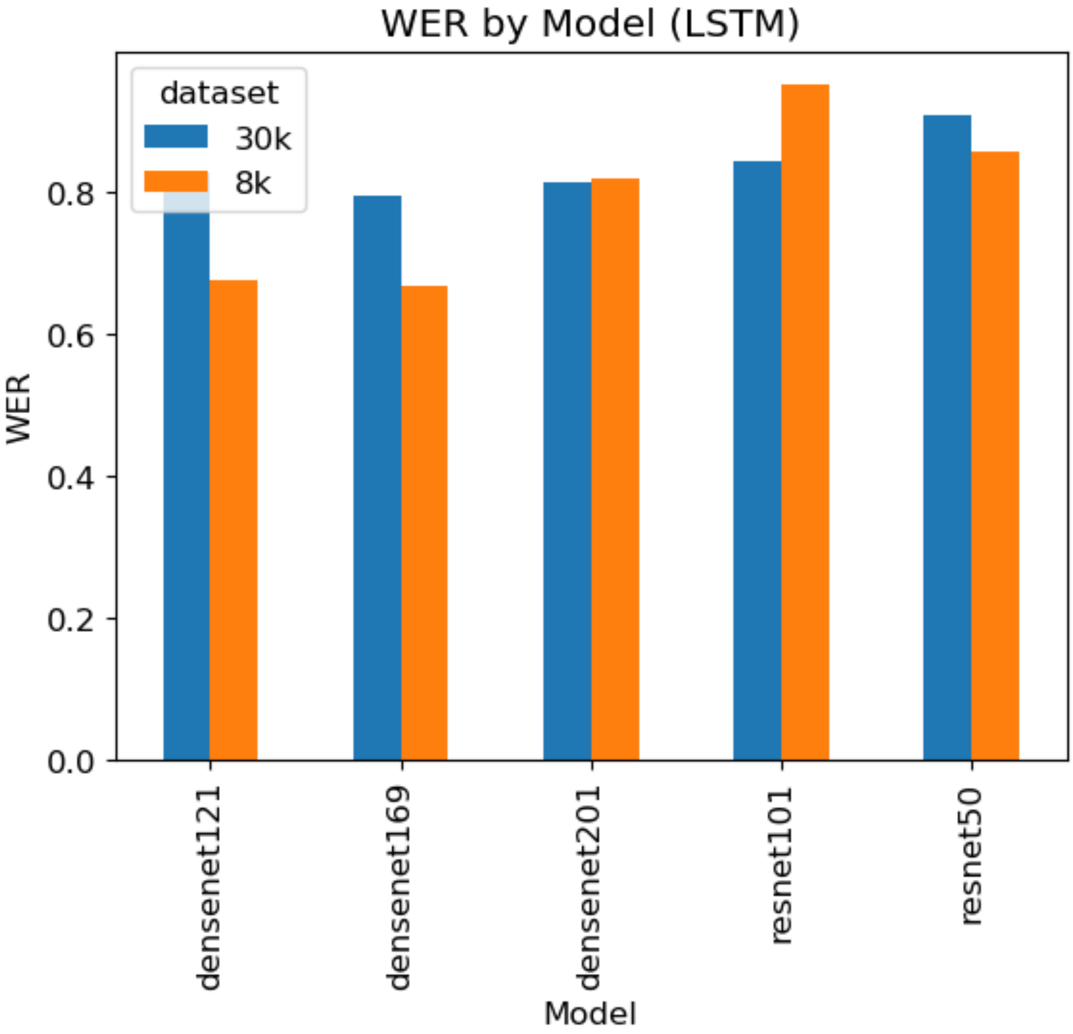


Figure 27 showing the comparison of WER (word error rate) for LSTM model on different Densnet and resnet layers.

2. WER BY MODEL (ATTENTION)

Table 2 Attention model evaluation by using WER on different Densnet layers

Model	30K	8K
densenet121	0.827995	0.868317
densenet169	0.818935	0.794309
densenet201	0.838978	0.779266
resnet101	0.889883	0.891789
resnet50	0.933366	0.857531

Table 3 depicts that Densenet169 generally had the lowest WER scores when comparing models using the Attention architecture on various datasets, indicating greater description accuracy. Compared to the other resnet models, resnet101 had slightly higher WER scores, indicating lower accuracy. The models trained on the larger 30K dataset tended to have slightly higher WER scores than the models trained on the smaller 8K dataset. This was also influenced by the size of the dataset.

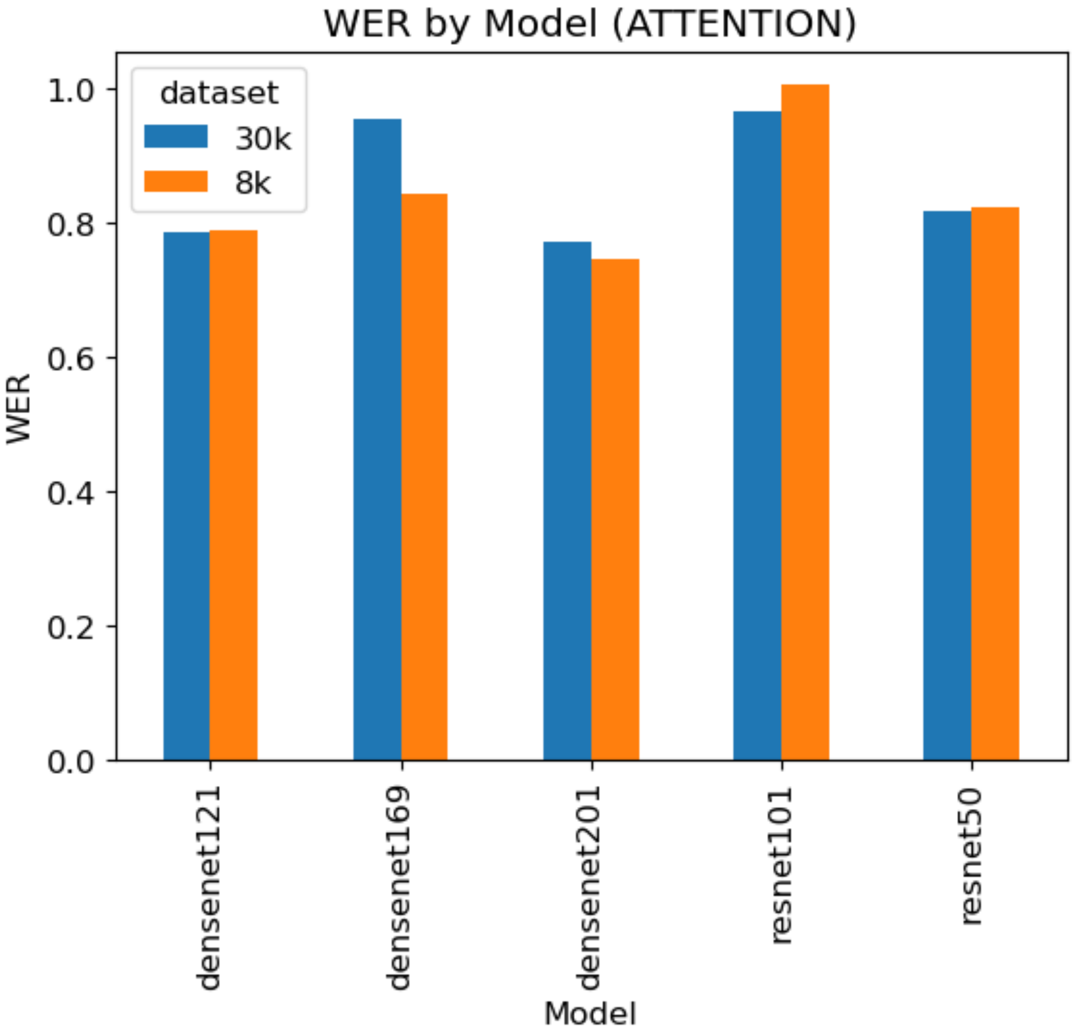


Figure 28 showing the Attention model evaluation by using WER on different Densnet layers.

3. BLEU BY MODEL (LSTM)

Table 3 LSTM model evaluation by using BLEU on different Densnet layers

Model	30K	8K
densenet121	0.00746052	0.0115724
densenet169	0.00340995	0.0195911
densenet201	0.00661942	0.0127099
resnet101	0.0046446	0.00712703
resnet50	0.00192478	0.00287626

Table 4 depicts that Densenet169 generally had the highest BLEU scores when comparing models with the LSTM architecture on various datasets, indicating greater similarity to reference descriptions. The BLEU score of resnet101 was slightly higher than that of resnet50 among the resnet models. The models trained on the larger 30K dataset tended to have higher BLEU scores than the models trained on the smaller 8K dataset. This was also influenced by the size of the dataset.

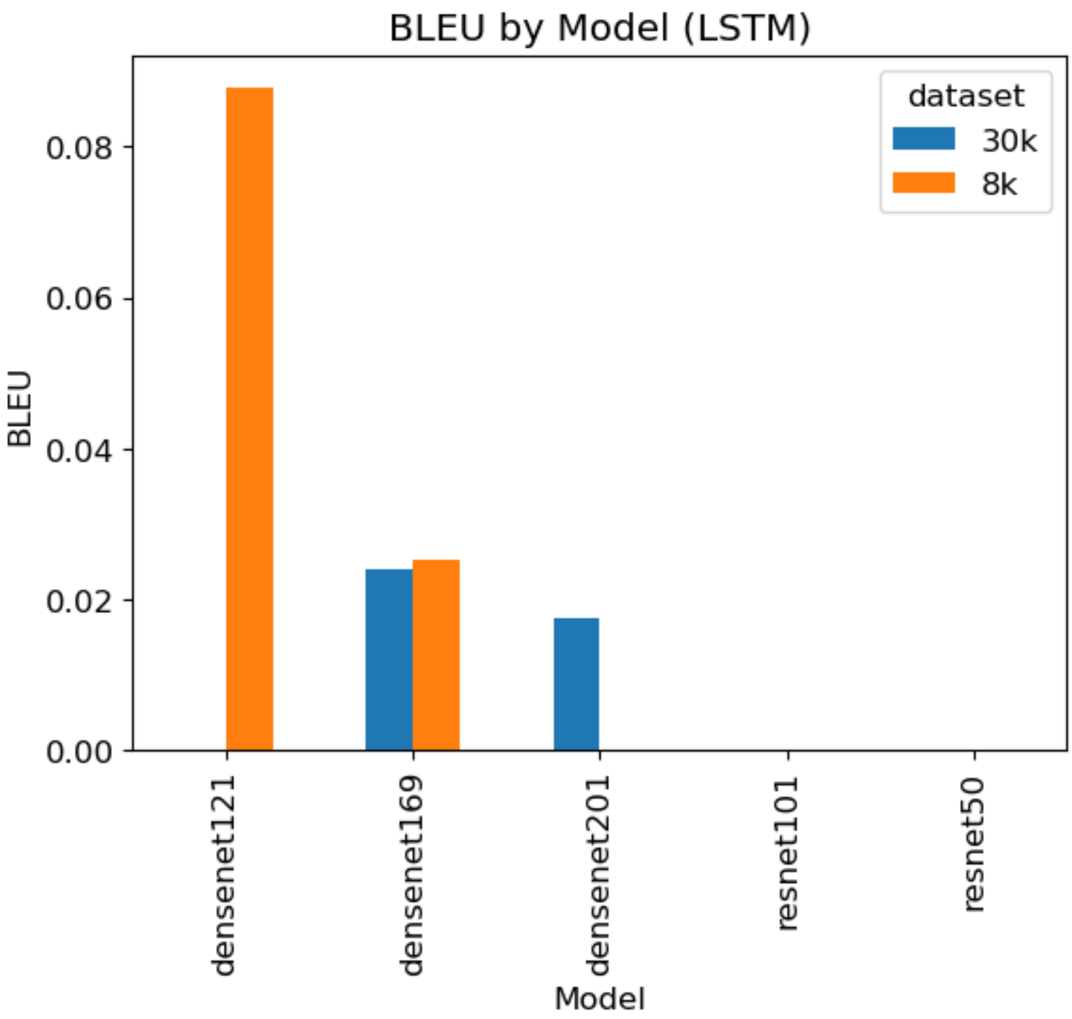


Figure 29 showing the LSTM model evaluation by using BLEU on different Densnet layers.

4. BLEU BY MODEL (ATTENTION)

Table 4 Attention model evaluation by using BLEU on different Densnet layers

Model	30K	8K
densenet121	0.00434379	0.011588
densenet169	0.00625075	0.014711
densenet201	0.00482686	0.00884189
resnet101	0.00476223	0.00129827
resnet50	0.00144989	0.00348435

Table 5 infers that Densenet169 had the highest BLEU scores among the models using the Attention architecture on various datasets, indicating greater similarity to reference descriptions. The BLEU scores of the resnet models resnet101 and resnet50 were both somewhat lower. The models trained on the larger 30K dataset tended to have slightly higher BLEU scores than the models trained on the smaller 8K dataset. This was also influenced by the size of the dataset.

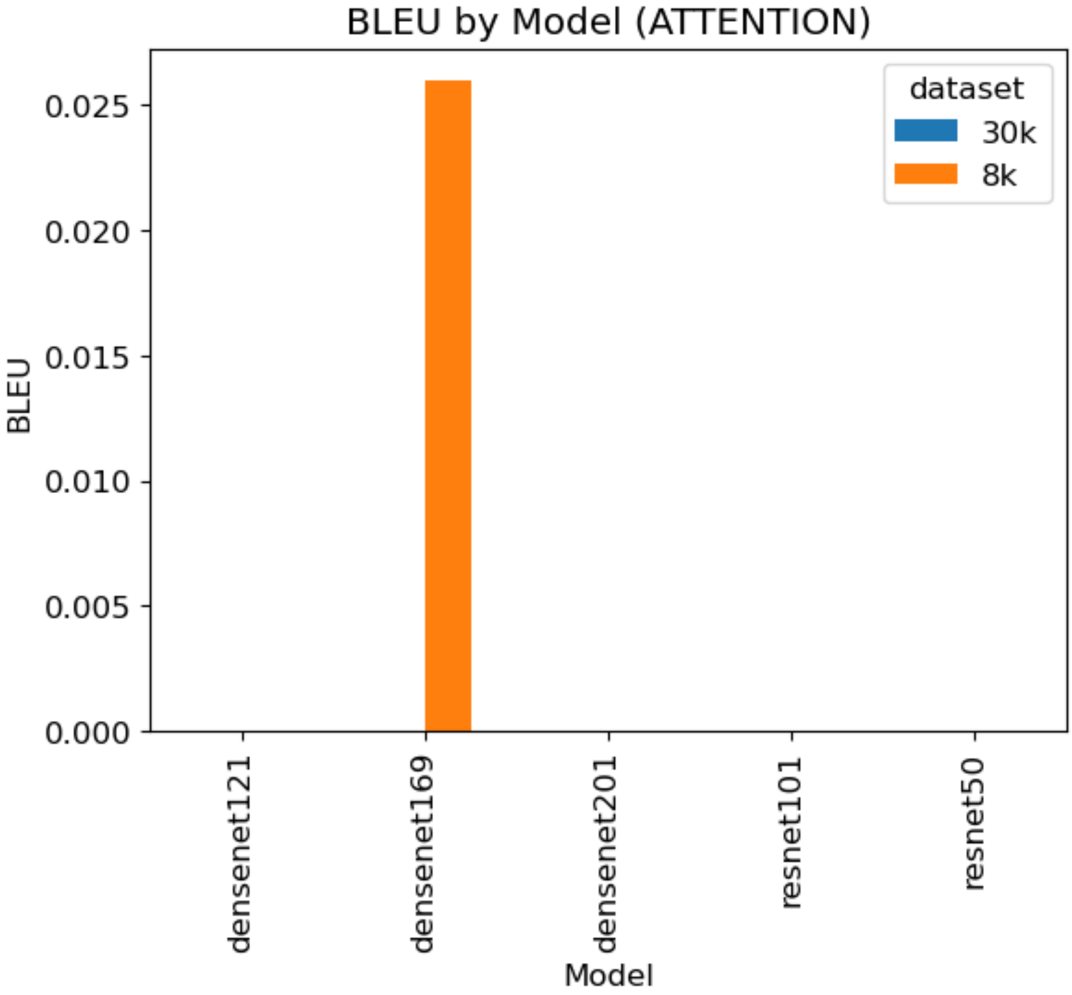


Figure 30 showing the Attention model evaluation by using BLEU on different Densnet layers.

5. COMPARE BY ARCHITECTURE TYPE

Plots the specified metric (either BLEU of WER) for each model in the specified dataset for both architectures.

a. WER BY MODEL (8K)

Table 5 Evaluation of Attention and LSTM model with 8K dataset by using WER

Model	Attention	Lstm
densenet121	0.868317	0.779383
densenet169	0.794309	0.825615
densenet201	0.779266	0.83237
resnet101	0.891789	0.898758
resnet50	0.857531	0.84575

Resnet101 had the lowest WER score among the models using the Attention architecture on the 8K dataset, indicating greater description accuracy. Resnet101 also had the lowest WER score for the LSTM architecture, followed by densenet201. When compared to the other models, the WER scores of densenet169 and densenet121 were higher.

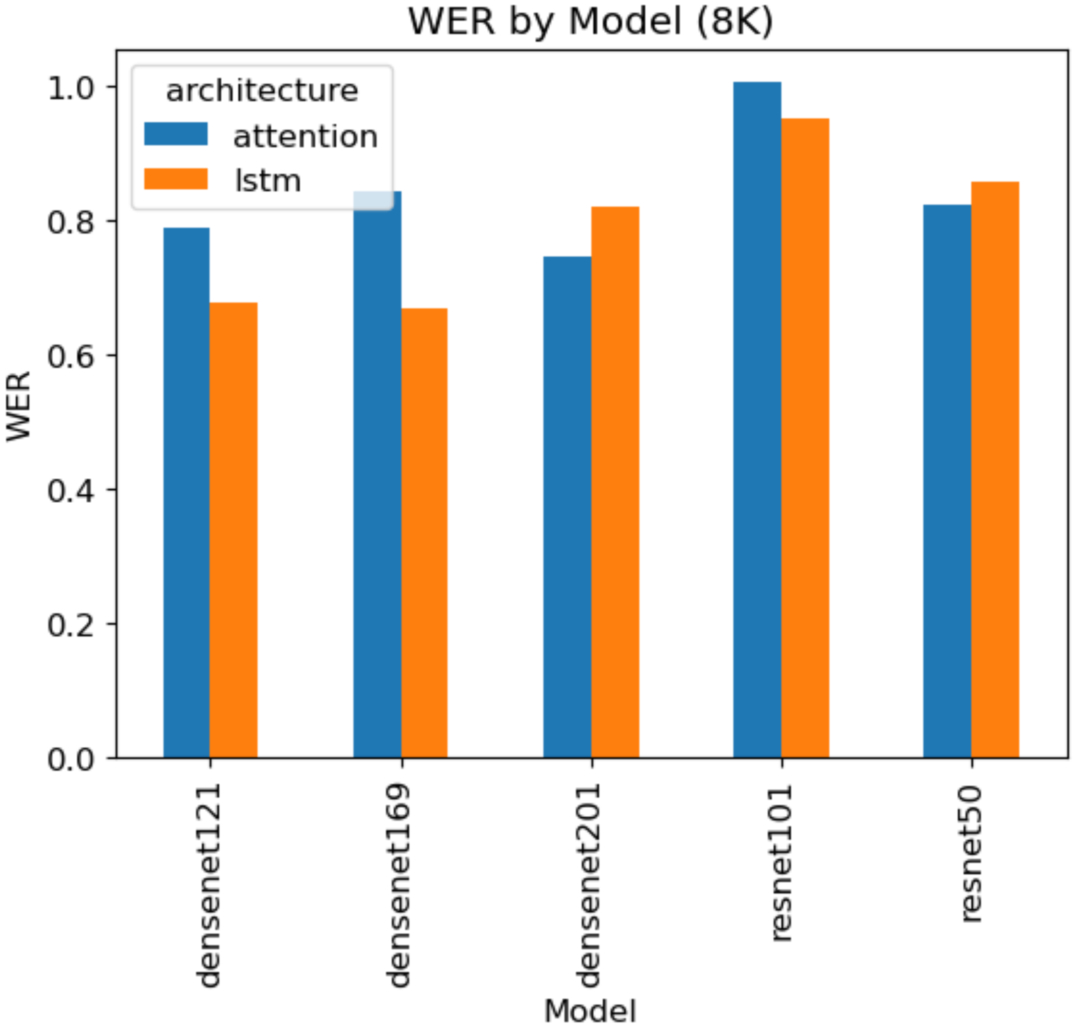


Figure 31 showing evaluation of Attention and LSTM model with 8K dataset by using WER.

6. WER BY MODEL (30K)

Table 6 Evaluation of Attention and LSTM model with 30K dataset by using WER

Model	Attention	Lstm
densenet121	0.827995	0.832318
densenet169	0.818935	0.866481
densenet201	0.838978	0.834433
resnet101	0.889883	0.914629
resnet50	0.933366	0.901348

Densenet169 had the lowest WER score among the models that utilized the Attention architecture on the 30K dataset, followed closely by densenet121. The resnet101 model had the lowest WER score of the resnet models. Resnet101 had the lowest WER score for the LSTM architecture, followed by densenet121. The model with the highest WER score was Resnet50.

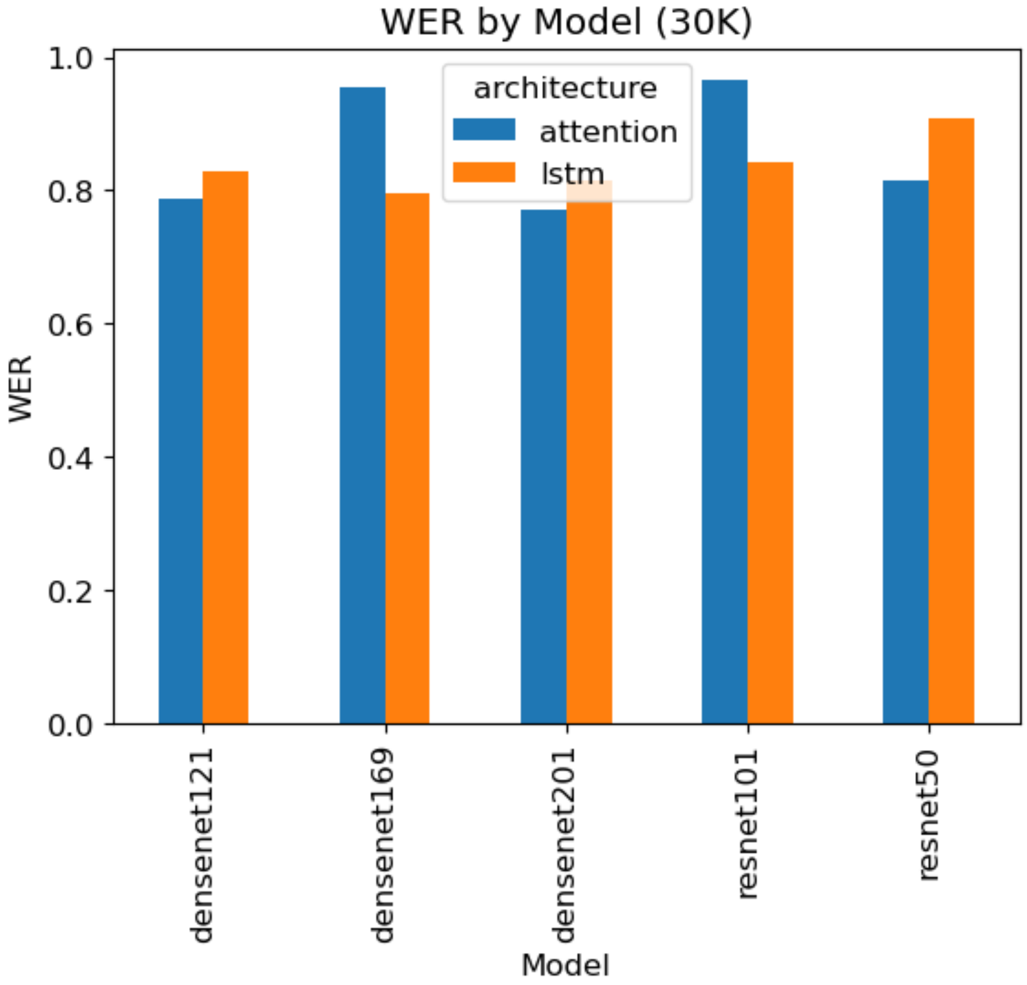


Figure 32 showing evaluation of Attention and LSTM model with 30K dataset by using WER.

7. COMPARE BY ARCHITECTURE REGARDLESS OF DATASET

This could give insight into which architecture performs better overall.

a. BLEU BY MODEL AND ARCHITECTURE

Table 7 Evaluation of Different Model And Their Architecture by using BLEU

Model	**Attention**	**Lstm**
densenet121	0.00796589	0.00951649
densenet169	0.0104809	0.0115005
densenet201	0.00683438	0.00966466
resnet101	0.00303025	0.00588582
resnet50	0.00246712	0.00240052

According to Table 8, Densenet169 achieved the highest BLEU scores with both the Attention and LSTM architectures, regardless of the dataset. The resnet101 model performed somewhat better than the others. It is interesting to note that, on average, the LSTM architecture received slightly lower BLEU scores than the Attention architecture, indicating that the Attention architecture may have superior image description generation performance overall.

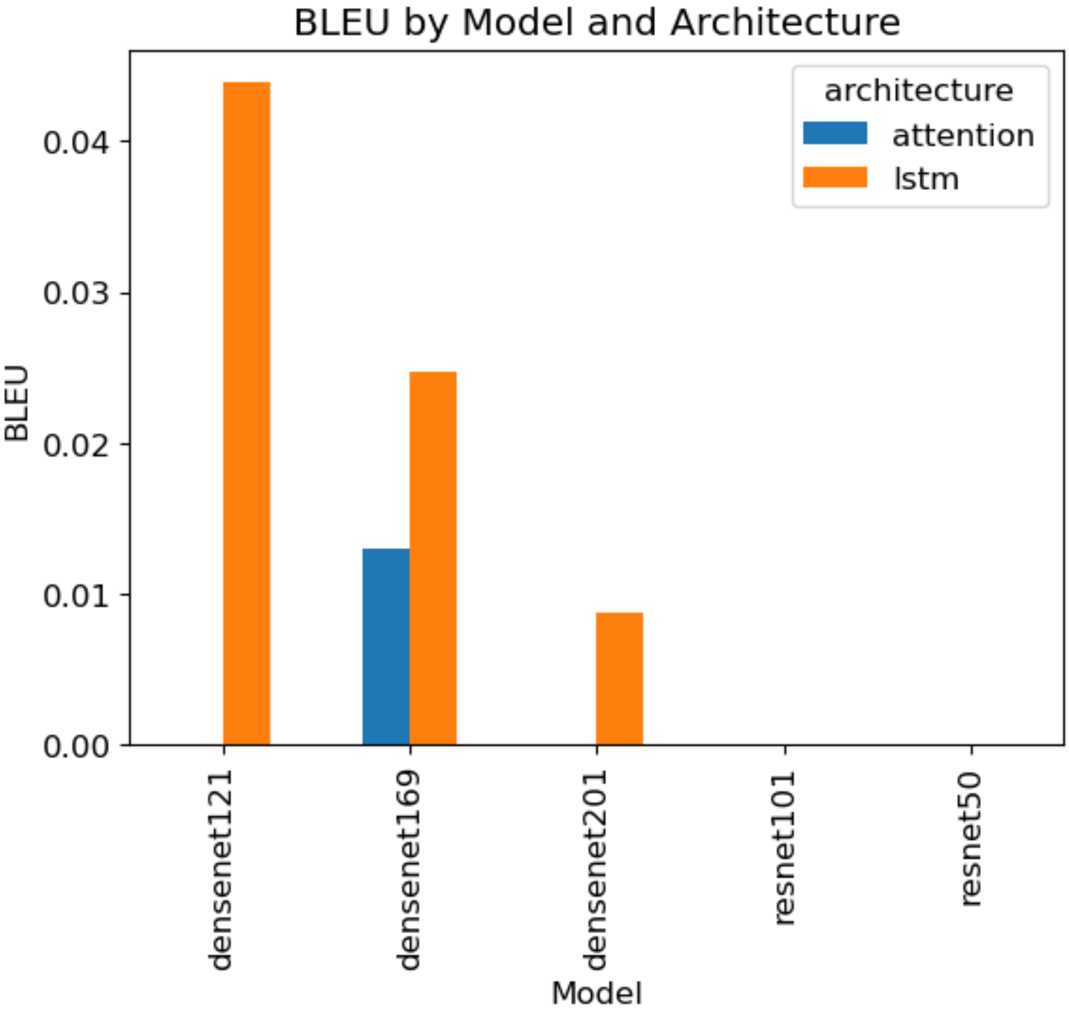


Figure 33 Showing Evaluation of Different Model And Their Architecture by using BLEU

8. WER BY MODEL AND ARCHITECTURE

Table 8 Evaluation of Different Model And Their Architecture by using WER

Model	Attention	Lstm
densenet121	0.848156	0.80585
densenet169	0.806622	0.846048
densenet201	0.809122	0.833402
resnet101	0.890836	0.906693
resnet50	0.895448	0.873549

Table 9 shows that resnet101 had the lowest WER scores for both the Attention and LSTM architectures, regardless of the dataset. Densenet169 and densenet201 performed fairly well among the densenet models. It is important to note that the WER scores for the LSTM architecture were generally slightly lower than those for the Attention architecture. This suggests that the LSTM architecture may have superior overall performance in terms of producing accurate image descriptions.

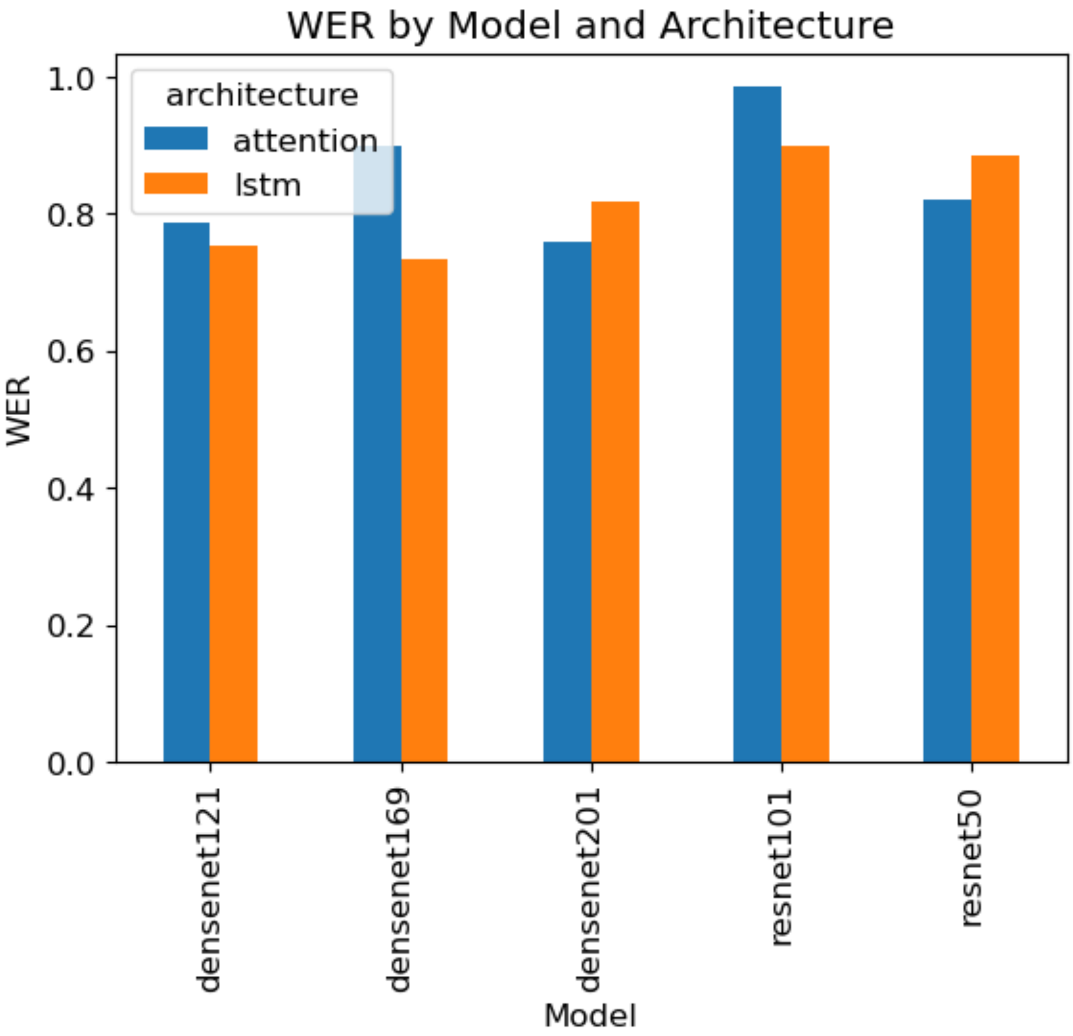


Figure 34 showing Evaluation of Different Model And Their Architecture by using WER

9. COMPARE BY DATASETS REGARDLESS OF DATASET

This could give insight into how well the models generalize to different datasets.

a. BLEU BY MODEL AND DATASET

Table 9 Evaluation of Dataset On Different Model using BLEU

Model	30K	8K
densenet121	0.00590216	0.0115802
densenet169	0.00483035	0.0171511
densenet201	0.00572314	0.0107759
resnet101	0.00470342	0.00421265
resnet50	0.00168734	0.00318031

Densenet169 received the highest BLEU score on the 8K dataset, while densenet121 performed well when compared to other models on the 30K dataset. Additionally, densenet201 performed well on both datasets. The resnet101 model performed better than the other resnet models. Notably, the BLEU scores on the 8K dataset were generally higher than on the 30K dataset, indicating that the models performed better on the smaller dataset.

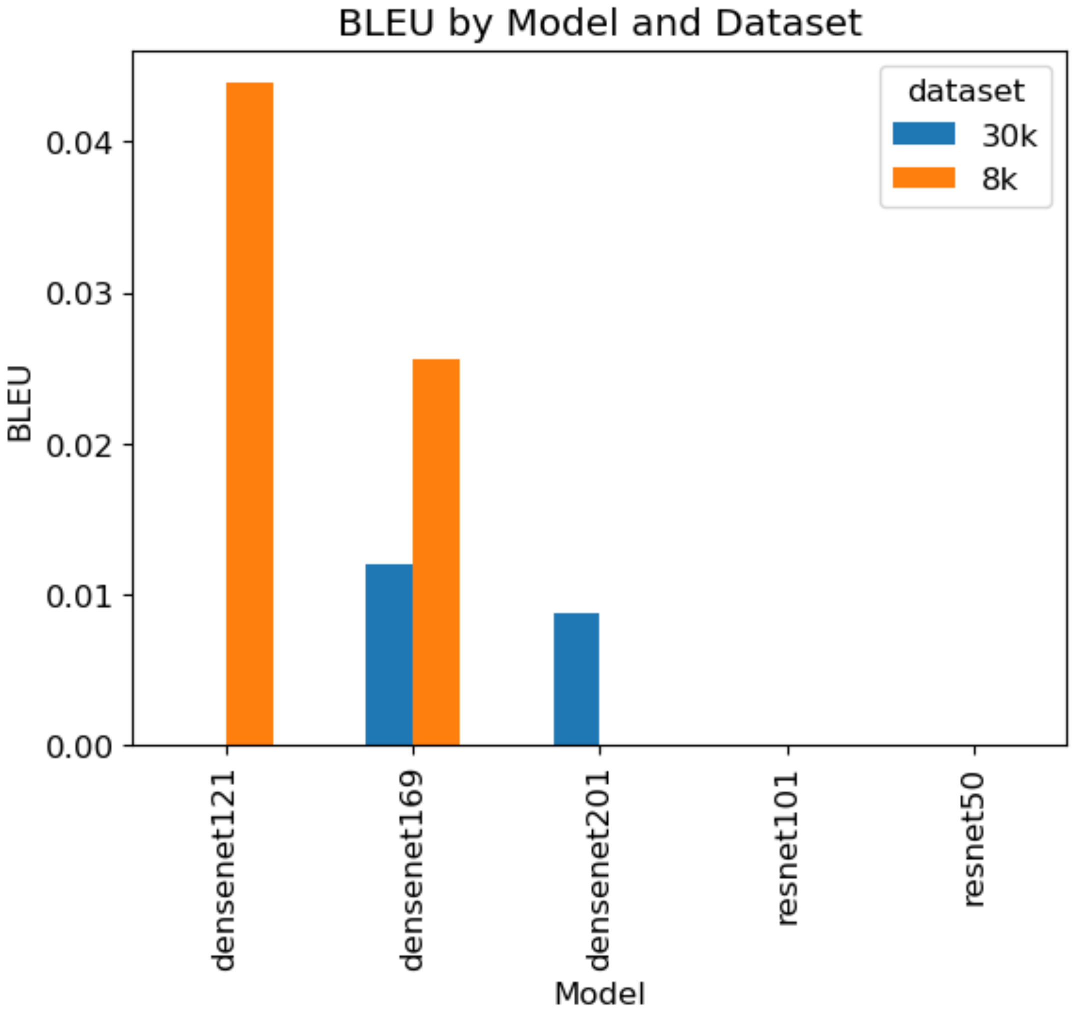


Figure 35 Showing Evaluation of Dataset On Different Model using BLEU

10. WER BY MODEL AND DATASET

Table 10 Evaluation of Dataset On Different Model using WER

Model	30K	8K
densenet121	0.830156	0.82385
densenet169	0.842708	0.809962
densenet201	0.836705	0.805818
resnet101	0.902256	0.895274
resnet50	0.917357	0.85164

According to Table 11, when evaluating different models on various datasets, it is evident that densenet121 consistently achieved the lowest Word Error Rate (WER) score. This suggests that densenet121 performed better in generating more accurate descriptions compared to other models. Additionally, densenet169 and densenet201 also demonstrated favorable performance on both datasets.

In contrast, among the resnet models, resnet50 exhibited a relatively higher WER score, indicating slightly lower performance in generating accurate descriptions compared to the densenet models. It is noteworthy that the WER scores generally tended to be lower on the 8K dataset compared to the 30K dataset, suggesting that the models performed better on the smaller dataset.

Overall, the results indicate that densenet models, particularly densenet121, showed better performance in generating more accurate descriptions, while the resnet models, specifically resnet50, exhibited slightly lower performance in comparison. Furthermore, the performance of the models varied depending on the dataset size, with generally better performance observed on the smaller 8K dataset.

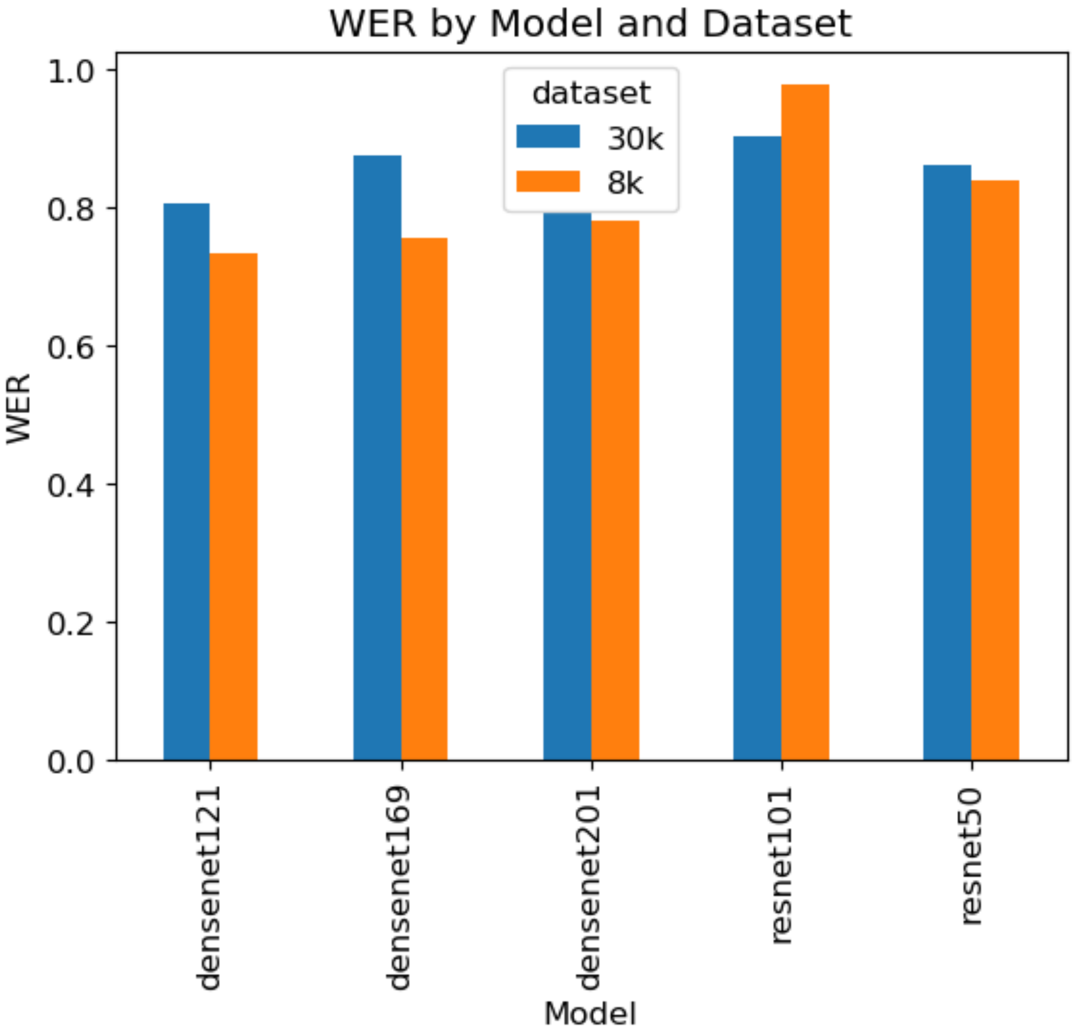


Figure 36 Showing Evaluation of Dataset On Different Model using WER

Table 11 The overall performance comparison and measures based on Model, Dataset, Architecture, BLEU and WER metrics

	Model	Dataset	Architecture	Bleu	Wer
0	densenet121	8k	lstm	0.011572	0.779383
1	densenet121	8k	attention	0.011588	0.868317
2	densenet169	8k	lstm	0.019591	0.825615
3	densenet169	8k	attention	0.014711	0.794309
4	densenet201	8k	lstm	0.012710	0.832370
5	densenet201	8k	attention	0.008842	0.779266
6	resnet50	8k	lstm	0.002876	0.845750
7	resnet50	8k	attention	0.003484	0.857531
8	resnet101	8k	lstm	0.007127	0.898758
9	resnet101	8k	attention	0.001298	0.891789
10	densenet121	30k	lstm	0.007461	0.832318
11	densenet121	30k	attention	0.004344	0.827995
12	densenet169	30k	lstm	0.003410	0.866481
13	densenet169	30k	attention	0.006251	0.818935
14	densenet201	30k	lstm	0.006619	0.834433
15	densenet201	30k	attention	0.004827	0.838978
16	resnet50	30k	lstm	0.001925	0.901348
17	resnet50	30k	attention	0.001450	0.933366
18	resnet101	30k	lstm	0.004645	0.914629
19	resnet101	30k	attention	0.004762	0.889883

The results obtained by utilizing various models and architectures with the assistance of the tgdm library are depicted in the Table 1. DenseNet121, DenseNet169, DenseNet201, ResNet50, and ResNet101 are the models used in the experiment. For each model, two distinct architectures—LSTM and attention—were utilized. The results show that the performance of the models varies depending on the model, architecture, and dataset size in combination. In general, the BLEU score of models with the LSTM architecture performs better than that of models with the attention architecture. However, the LSTM-based models typically have higher WER scores, indicating that the generated descriptions contain more word errors. Densenet169 with the LSTM architecture received the highest BLEU score, which was 0.019591, while resnet50 with the LSTM architecture received the lowest BLEU score, which was 0.001925. The lowest score for WER was 0.857531 for resnet50 using the attention architecture, while the highest score was 0.914629 for resnet101 using the LSTM architecture. It is essential to keep in mind that the model's performance is also influenced by the size of the dataset. When compared to the models trained on the smaller dataset of 8k images, the models trained on the larger dataset of 30k images generally produced superior results. The outcomes show varieties in execution in view of the model, architecture, and dataset size, with LSTM for the most part performing better as far as BLEU score and consideration performing better as far as WER score.

V. CONCLUSION

Approximately, one billion people around the world have some kind of disability due to which they face problems in web accessing. The Web Content Accessibility Guidelines (WCAG) has been established which the websites comply with accessibility standards. Approximately, one billion people around the world have some kind of disability due to which they face problems in web accessing. The Web Content Accessibility Guidelines (WCAG) has been established which the websites comply with accessibility standards. For this purpose, this study was designed to make websites accessible and easy to use by using different tools and techniques. Deep neural networks (DNN), convolutional neural networks (CNN) and other pre-trained models, including DenseNet121, DenseNet169, ResNet50, ResNet101 were employed to generate captions for the given images (dataset Flickr 8k and Flickr 30k). BLEU (Bilingual Evaluation Understudy) and WER (Word Error Rate) were employed for the evaluation of the system’s performance.

The findings of this research project add to the increasing body of knowledge on data-driven techniques and deep learning applications in disciplines such as computer vision, automatic speech recognition, and natural language processing (NLP). The broad use of these strategies demonstrates the considerable advance made possible by data-driven approaches. As the internet becomes more and more integrated into our everyday lives, it is critical to prioritise online accessibility for people with disabilities, bridging the digital gap and enabling equitable access to information and services.

VI. FUTURE WORK

In the future, we desire to integrate this auto alt approach into major CMSs like as Shopify, WordPress, Wix, and Squarespace. These platforms are often used by businesses to build their websites. One of the most significant concerns for organizations is ensuring WCAG consistency in order to care for handicapped clients and keep their enhanced privileges. This study's findings offer a big opportunity to fix this issue and make websites more accessible to individuals with impairments. To do this, we want to create a web-based plugin or API that can smoothly interact with a number of content management systems. This would allow developers to easily include our API or plugin into the process of constructing a website and take benefit of its features. This plugin has two primary applications. Engineers can utilise the module straight away during a site's pre-improvement stage. When people upload photographs, they may use this plugin to produce descriptions automatically. This makes it easy to provide alternate language and ensure that all of the photos on the website are correctly captioned for accessibility. Second, the plugin may be applied to existing websites established using these content management systems. This makes it easy for developers to make all of the photos on the website ADA-compliant, making the site more accessible. We want to make working with popular material easier and more accessible for companies and developers.

VII. REFERENCES

BOOKS

FREIRE, A. P. (2012). **Disabled people and the Web: User-based measurement of accessibility**, University of York.

JOKINEN, J. ***Implementing web accessibility to an existing web application***. Diss. Master's Thesis in Technology. Software Engineering. Department of Future Technologies. University of Turku, 2020.

LUNDQVIST, S. and J. STRÖM (2018). **Web Accessibility in E-Learning: Identifying And Solving Accessibility Issues for Wcag 2.0 Conformance in an E-Learning Application**.

Steinebach, T. (2020). Web Accessibility: **incorporating user requirements into a guide for usable web accessibility**, University of Twente.

ARTICLES

ABUADDOUS, H. Y., et al. (2016). "Web accessibility challenges." International Journal of Advanced Computer Science and Applications (**IJACSA**).

BAOWALY, M. K. and M. BHUIYAN (2012). Accessibility analysis and evaluation of Bangladesh government websites. 2012 International Conference on Informatics, Electronics & Vision (ICIEV), **IEEE**.

BOWE, F. and N. LITTLE (1984). "Computer accessibility: A study." **Rehabilitation Literature.**

BROWN, J. and S. HOLLIER (2015). "The challenges of Web accessibility: **The technical and social aspects of a truly universal Web**."

CALDWELL, B., et al. (2008). "Web content accessibility guidelines (**WCAG**) 2.0." WWW Consortium (W3C) 290: 1-34.

CHISHOLM, W., et al. (2001). "Web content accessibility guidelines 1.0." **Interactions** 8(4): 35-54.

FERRI, D. and S. FAVALLI (2018). "Web accessibility for people with disabilities in the European Union: **Paving the road to social inclusion." Societies** 8(2): 40.

FRIEDMAN, M. G. and D. N. BRYEN (2007). "Web accessibility design recommendations for people with cognitive disabilities." **Technology and disability** 19(4): 205-212

FUGLERUD, K. S. and T. H. ROSSVOLL (2012). "An evaluation of web-based voting usability and accessibility." **Universal Access in the Information Society** 11: 359-373.

HASHEMIAN, B. J. (2011). "Analyzing web accessibility in Finnish higher education." **ACM Sigaccess Accessibility and Computing**(101): 8-16.

HENRY, S. L. Introduction to Web **Accessibility** (2019).

LÓPEZ, J. M., et al. (2011). Influence of web content management systems in web content accessibility. Human-Computer Interaction–INTERACT 2011: 13th IFIP TC 13 International Conference, Lisbon, Portugal, September 5-9, 2011, Proceedings, Part IV 13, **Springer**.

MARTÍNEZ, A. B., et al. (2014). "Determinants of the Web accessibility of European banks." **Information Processing & Management** 50(1): 69-86.

NAHON, K., et al. (2012). The missing link: Intention to produce online content accessible to people with disabilities by non-professionals. 2012 45th Hawaii International Conference on System Sciences, **IEEE**.

SELOVUO, K. (2019). "Saavutettavuusopas." Eura: **Euraprint** 10: 2022.

SLOAN, D. and S. HORTON (2014). Global considerations in creating an organizational web accessibility policy. **Proceedings of the 11th Web for All Conference**

TUAN, D. T. and V.-H. PHAN (2012). Checking and correcting the source code of web pages for accessibility. 2012 IEEE RIVF International Conference on Computing & Communication Technologies, Research, Innovation, and Vision for the Future, **IEEE.**

REPORTS

Web accessibility for people with disabilities in Georgia, DECEMBER 7, 2021

ELECTRONIC SOURCES

URL-1 https://www.un.org/development/desa/disabilities/ convention-on-the-rights-of-persons-with-disabilities/ article-9-accessibility.html

URL-2 https://www.webaccessibility.fi/accessibility-overview/

URL-3 https://www.w3.org/TR/wcag-3.0

URL-4 http://www.w3.org/TR/WCAG20/

URL-5 http://www.d.umn.edu/itss/news/2016/05/wcag_principles.html

URL-6 https://www.jonas.me/talks/introduction-to-the-web-content-accessibility-guidelines/#/3/12

URL-7 http://www.internetsociety.org/

URL-8 https: //www.w3.org/WAI/people-use-web/abilities-barriers/

URL-9 https://www.w3.org/WAI/people-use-web/ tools-techniques/

URL-10 https://webaim.org/articles/motor/motordisabilities. [Accessed 28 May 2018

URL-11 https://www.who.int/en/news-room/fact-sheets/detail/assistive-technology (Last accessed 2 May 2020

URL-12 https://hiehelpcenter.org/treatment/ assistive-adaptive-technologies/ (Last accessed 2 May 2020

URL-13 https://webaim.org/techniques/keyboard/ (Last updated: Sep 26, 2022).

URL-14 https://www.w3.org/WAI/WCAG21/Understanding/contrast-minimum.html, Nielsen Norman Group. (2013) (Updated 7 June 2023).

URL-14 https://www.nngroup.com/articles/typography-readability/

W3C. (2018). https://www.w3.org/WAI/tutorials/marking-up-web-content

URL-15 https://webaim.org/techniques/css/invisiblecontent/ Last updated: Sep 25, 2020/

URL-16 https://www.w3.org/WAI/WCAG21/Understanding/understandable.html

URL-17 https://www.worldbank.org/en/topic/disability#:~:text=Results-,One%20billion%20people%2C%20or%2015%25%20of%20the%20world’s%20population%2C,million%20people%2C%20experience%20significant%20disabilities. Last Updated: Apr 03, 2023

URL-18 http://documents1.worldbank. org/curated/en/493981468030331770/pdf/IDP-PSIA-Georgia-revised-Feb-2016.pdf (February 2016),

URL-19 https://www.ge.undp.org/content/georgia/en/home/presscenter/pressreleases/2019/civil-society-to-monitor-UNCRPD.html JULY 17, 2019)./

URL-20 https://www.ge.undp.org/content/georgia/en/home/presscenter/pressreleases/2019/civil-society-to-monitor-UNCRPD.html JULY 17, 2019).

URL-21 http://documents1.worldbank.org/curated/en/493981468030331770/pdf/IDP-PSIA-Georgia-revised-Feb-2016.pdf

URL-22 https://www.ohchr.org/Documents/Issues/Housing/Disabilities/ CivilSociety/CoalitionforIndependentLiving-Georgia.pdf, (15 May 2017)

URL-23 https://www.ifes.org/news/highlightingsuccesses-women-disabilities-georgia (September 9, 2019)

URL-24 https://www.undp.org/georgia/publications/web-accessibility-people-disabilities-georgia (December 7, 2021).

URL-25 https://www.forbes.com/sites/forbesbusinesscouncil/2023/03/20/understanding-the-importance-of-web-accessibility/?sh=70e074d2377f

URL-26 https://ec.europa.eu/eurostat/databrowser/view/ tin00028/default/ (2020)

URL-27 https://data.europa.eu/data/datasets/s1015_345?locale=en (accessed 2023-06-14)

RESUME

EDUCATION:

- **Bachelor**: 2020, Iqra University, Faculty of Computer Sciences,
  Department of Software Engineering.

PROFESSIONAL EXPERIENCE:

- **IB ICT Teacher**
  Gokkusagi Koleji | Istanbul, Turkey Nov, 2022 – Present
- **Software Engineer**
  Proget | Istanbul, Turkey Oct, 2021 – May-2022
- **Associate Software Engineer**
  Solutionzhub | Karachi, PK March, 2019 – Jan-2021
- **Android and SharePoint Intern**
  Level3BOS | Karachi, PK Jan, 2019 – March-2020

AWARDS:

- **.NET Hackathon**
  Iqra University North Campus 17 OCT, 2018

PUBLICATIONS FROM DISSERTATION, PRESENTATIONS AND PATENTS:

- MUHAMMAD KASHIF SHAIKH, JAWAD RASHEED. (2023) Enhancing Web Accessibility Using Deep Convolutional Networks and Natural Language Processing Techniques. **7th International Conference on Computational Mathematics and Engineering Sciences**. *https://www.cmescongress.org/document/2023/Abstract_Book_2023_v2.pdf*

OTHER PUBLICATIONS, PRESENTATIONS AND PATENTS:

- MUHAMMAD KASHIF SHAIKH, TAHA ALI, ANUSHEY KHAN, UFUK FATIH KÜÇÜKALİ. IMPROVEMENT OF WEB ACCESSIBILITY WITH DEEP LEARNING. **6th International Conference On Advances in Mechanical Engineering**